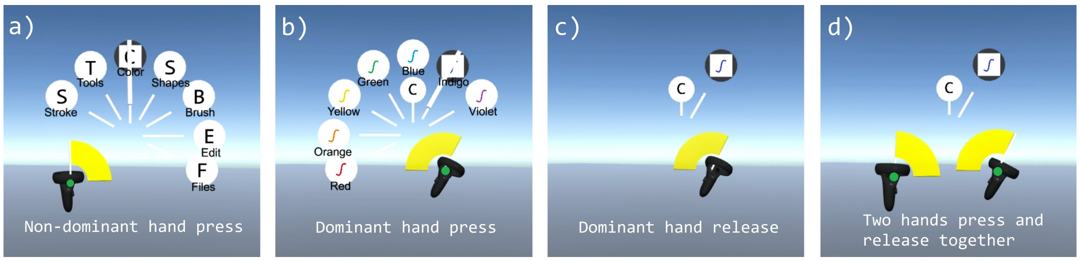
# VRLin: Proprioceptive Rotational Menus for HMD Command Access

ANONYMOUS AUTHOR(S)*




Users of Head Mounted Displays (HMDs) frequently need to access a rich command set to control applications and environments. However, current menu systems on HMDs typically require careful visual targeting, which can prove distracting when immersed in gaming or productivity on these devices. In this paper, we present VRLin, a symmetric, rotational menu suitable for HMD command access. By rotating controllers, users can access a rich command hierarchy of arbitrary depth. VRLin implements the principle of rehearsal-based learning by supporting both eyes-engaged command access (via visual feedback) and eyes-free command access (via proprioceptive targeting of command location). Through two formative studies, we determine appropriate parameters for both eyes-engaged and proprioceptive command access. A final, first-use assessment elicits user feedback on the efficacy of this command invocation technique for HMDs.

CCS Concepts: • **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Command selection, Virtual Reality, Rehearsal-based learning.

**ACM Reference Format:**

Anonymous Author(s). 2018. VRLin: Proprioceptive Rotational Menus for HMD Command Access. In *CHI '21: The ACM CHI Conference on Human Factors in Computing Systems, May 08–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/1122445.1122456

## 1 INTRODUCTION

Alongside the input of data or text [37], one frequent target of research innovation is command selection [36]. Alongside menus and buttons [10], researchers have explored how best to access command shortcuts [27], how to design command gestures [34, 36], and how to leverage rehearsal [17, 22] and cross-modal learning [18, 40] to support rich, rapid command selection.

Head-Mounted Displays (HMDs) that create virtual reality (VR) environments present unique challenges for command selection. Because the user is visually engaged with an immersive scene in which they are embedded, we would argue that, as in, e.g., computer displays [22], SmartTvs [40], and Head-Mounted Displays (HMDs) [37], there is a benefit to a user not needing to view and interact with a user interface, i.e. to support command input in an "eyes-free" manner [40]. Researchers have long suggested that the display of a user interface introduces visual disruption and slows user input due to the time taken to display the interface, the time taken to process the visual disruption of the display, and the time taken to visually acquire targets [22, 27]. To eliminate the cost of interface display, researchers have looked at various techniques to support input without the need to display an interface [18, 27, 40]. One of the most common techniques to support input in the absence of a displayed interface is *rehearsal-based learning*, a technique for skilled command selection first introduced by Kurtenbach in his work on Marking Menus [22]. Various researchers have employed rehearsal-based learning to support eyes-free command selection [6, 16].

In this paper, we introduce VRLin (see Figure ), a command selection technique that leverages wrist rotation to support rehearsal-based input in VR systems rendered with HMDs. Via the sequential selection of angular pie menu segments, a rich command set can be displayed to the end user. Through rehearsal, the end-user can move from visual-search-and-select menu input to proprioceptive angle matching, with the goal that command selection naturally moves from visual targeting of menu items to, through learning, expert command activation via muscular memory, without visual display of an interface. VRLin is possible in both unimanual and bimanual variants.

To create VRLin, this paper describes three studies: a pilot study that assesses typical wrist rotation ranges when leveraging a VR Controller; a formative assessment of rotational precision that demonstrates that users can accurately target up to seven rotational pie segments using proprioception alone (i.e. "eyes-free") with both dominant and non-dominant hand; and a design review study that assesses VRLin's unimanual and bimanual variants against a traditional pointing-based menu selection technique and a non-dominant hand palette-based command selection technique [35]. In our design review study, we found that users could accurately input commands in VRLin, that, with minimal training, they achieved high accuracy, and that early design review from users familiar with VR-based command selection argue for the potential of this technique.

The remainder of this paper is organized as follows. We first examine related work in VR-based command selection, related command selection techniques focused on those that leverage rotation, and rehearsal-based learning. Next, we describe both the design of VRLin and results from our formative and first use studies. We conclude by arguing that VRLin represents an effective mechanism to support proprioceptive command selection in immersive virtual environments.

## 2 RELATED WORK

### 2.1 VR Command Selection Research

Alongside direct manipulation of the environment, a common need in virtual environments involves selecting commands. In modern commercial systems, it is common for VR-based systems to support command selection by displaying a command palette in front of the user and allowing the user to target individual widgets, a type of floating toolbar. For example, tutorials on creating user interfaces using Unity leverage the UIHelper prefab to support ray casting to control pointer position coupled with UI Canvases to render clickable widgets.

By far the most common command selection technique in commercial VR application design leverages the linear panel menus described in the aforementioned tutorial, but other styles of interaction have been explored over the past few decades [13]. Specifically considering pie menus, the menu design adopted by VRLin, there are several variants of hierarchical pie menus for 3D virtual environments [14, 21, 32] . Azai et al. introduced a on-body hierarchical menu in mixed reality [5], where menus are projected on the forearm. They include a rotation-based interaction; however, their rotation-based interaction involves using one hand to rotate the menu on the other arm, as one would rotate a bracelet. Monteiro et al. explored the performance of radial and panel menus with two placement (fixed on wall or on hand) in VR and found that a traditional panel menu with a fixed wall placement performs better than others; their technique employed a round touchpad for radial selection [28]. Santos et al discovered that selection time on radial menus is faster than linear menus with raycast interaction, where one uses a directional mark (as in marking menus [22]) to access menu items. While many such techniques exist, none specifically explore the effectiveness of sequential unimanual and bimanual wrist rotation to hierarchically arrange and access commands.

### 2.2 Related Bi-manual and Rotational Command Selection

Alongside the above techniques, there exists work in bi-manual interaction, both asynchronous bi-manual interaction [15] and synchronous bi-manual interaction [24]. As one example recent work on asymmetric bi-manual gestural interaction leverages left hand gestures to determine input mode and right hand actions to point/provide input [25]. As well, wrist rotation has been explored more generally in human motor control research, and it is known that rotational targeting rotational wrist positions follows Fitts's Law (confirmed initially by [12] with hand unencumbered and replicated by [11]). Wrist rotation has been proposed in mobile interaction as a motion gesture to control a smartwatch [11], an interaction leveraged by several other researchers in designing systems for interaction within two-dimensional and three-dimensional spaces (e.g. [20, 31]. Finally, in the domain of smartphone input, wrist rotation has been used to define a motion gesture that can act as a gestural delimiter [34] and wrist rotation motion gestures have been commercially adopted by Motorolla smartphones as the *MotoAction* to invoke the camera app. As well, the TWuiST system [29] uses proprioceptive displacement of a smartphone, including wrist rotation, for smartphone-based interaction, and Crossnan et al. [11] studies the use of wrist rotation for command selection in sitting and walking mobile contexts.

### 2.3 Rehearsal-based Interfaces

As user's develop expertise in command access via menu systems, users move from behavior characterized by reading commands to remembering where commands are and moving semi-automatically toward the commands' recalled position [10]. This process, rehearsal-based learning [10, 17, 22, 27], has been frequently leveraged in interfaces to support novice-to-practiced behavioral transitions. Furthermore, if interfaces can be designed such that users can physically locate commands via proprioception [16, 22, 33], users can access commands without any need to visually display menus or command widgets.

A large body of research has been done in Rehearsal-based Interfaces. Marking menus and their derivatives [7, 22, 38, 39] use the angle of strokes to invoke commands. Rehearsal-based input has also been explored in the context of touch-based input. As one example, Gutwin et al. [16] (on touch tablets) and Lafreniere et al. [23] (on touchscreen watches) designed a rapid command selection technique (FastTap) which allows users to select commands with a single quick thumb-and-finger tap. After learning the spatial location of items an expert mode was introduced to support eyes-free interaction. Rehearsal-based learning has also been used in hand pose input to support expert level input. As one example, Bailly et al. introduced a bimanual command selection technique using finger count of two hands to select commands on two-level menus [6, 8]. Finally, visual feedback mechanisms have been used to teach users to perform gestures more accurately by guiding them toward the correct movements, another form of rehearsal-based training [19].

### 2.4 Eyes-Free Input

The term "eyes-free" is one that is overloaded in the research literature. In some cases, "eyes-free" implies that no UI is displayed for the user [22]. In others, interface and/or feedback elements are displayed, but not on the input device [40]. In both cases, there exists benefit for the user – not needing to visually acquire an interface can preserve immersion in application interaction, can reduce visual disruption, and can avoid costly video updates [22], and not needing to switch focus between screens can preserve user focus on their primary task [30, 37].

In our work, we primarily adopt the first definition of "eyes-free", i.e. input where the interface – in our case a command selection palette – is accessed by the user without being visually instantiated on the display. To achieve this level of input, users must leverage proprioceptive mechanisms to perform appropriate actions [22]. This proprioceptive input must be trained, but, once trained, can support rapid, accurate command invocation [22, 23, 27].

## 3 VRLIN

### 3.1 Motivation

Our initial motivation to explore sequential, rotation-based command access in VR was based upon what we perceive to be, based on our exploration of related work, an obvious and significant gap in VR-based command access. Because VR leverages hand-held controllers, and because of the scale of the virtual environment, command selection using panel menus means that these menus must be placed in the visual world. The designer is presented with two options: make them large and/or proximal to the user, speeding targeting via Fitts's Law but hiding a significant amount of content, or shrink them within the environment to preserve visual immersion, but at the cost of more difficult targeting. This initial observation gave rise to our interest in rotation-based menus, because, even in the case where menus are displayed visually to the user, it is angular tolerance that defines complexity of the targeting task [12]. Therefore, the menus can be "visually small" within the VR environment – to preserve, as much as possible, the visual immersion within the space – without increasing the "difficulty" of the targeting task. Furthermore, as users develop expertise, because users are aware of the relative rotation of their wrist, commands can be invoked without visual feedback, using proprioception, allowing the menus to shrink further or, even, vanish entirely, thus avoiding visual disruption of the virtual environment [22].

In our initial explorations of command selection in VR, one aspect that we noted for many VR-based command selection techniques is that command selection was almost universally uni-manual. Even bi-manual techniques [9] typically used one hand, the non-dominant hand, as a marker for command selection, i.e. a spatial referent to position the palette, and then subsequent command selection actions are performed via the dominant hand. However, based on our informal observations, it is also the case that command selection rarely interleaves, in time, with another user action. Users, always in our observations, select commands, and then perform actions. Given the presence of two-controllers, one additional question we posed was whether there might exist advantages to bi-manual command selection [24].

It was these initial observations that motivated our research question within VR-based command selection: Could we design a symmetric, proprioceptive command selection mechanism in VR? Should it be uni-manual or bi-manual? What range of rotation would be comfortable? How many commands could be supported? With these questions in mind, and inspired by past research in marking menus and rehearsal-based interfaces, we designed VRLin.

### 3.2 Description

VRLin is a multi-level, sequential, rotation-based, in-air menu system that supports proprioceptive feedback and rehearsal-based learning. Figure  demonstrates how VRLin implements these principles. In this Figure, we see the bimanual variant of VRLin; the unimanual variant is similar, but leverages only one hand to perform command selections.

While multi-level, sequentially-activated, and rotational command selection approaches have all been explored in VR, to the best of our knowledge their combination, as realized in VRLin, is unique. To select a command in VRLin, users rotate the controller to a specific angle, then select that angle. Potentially (bimanual) switching hands, they perform the same rotation-based command selection to activate a sub-menu command. Note that, while the act of selecting is sequential, in the bimanual condition participants are free to parallelize the rotation of non-dominant and dominant hand to perform menu -> sub-menu selections. Menus can be of arbitrary depth.

Alongside command activation, VRLin's menu layout preserves spatial positioning of menus and submenus. Thus, as in marking menus, users begin to gradually learn the placement of the menus, i.e. the rotation angle that is required to select commands.

While novice users require the visual representation of the menu, as users acquire expertise through rehearsal-based learning [22], they begin to memorize the physical rotation of controllers needed to activate commands. To limit visual disruption and GPU cycles required for menu display [17, 22], VRLin's visual presentation of command selection exists in two modes: *menu mode* and *auto mode*. Menu mode is geared toward users who are not yet expert with the system. Users press an assigned button on the controller and a pie menu appears in air about 1 meter from the users' eyes, parallel with their field of view. They rotate the wrist to the ideal angle using visual feedback and release the button to select on this level. Depending on the number of levels of the menu they wish to access, they continue to rotate and release to select the item. There is a delay of 250ms [22] for the menu of each level to be visible after pressing.

When users are familiar with some items, they can rotate quickly without visual feedback, which is the *auto*nomic mode of this technique. Auto mode and menu mode can be interleaved between menu levels; for example, when the user is familiar with where to fast select in the first level, they can quickly press and release without waiting for the visual feedback, then use menu mode to carefully go through the items on the second level and select from among them.

Given the potential of bimanual parallelization for VRLin's bimanual variant, two add additional forgiveness options were implemented in bi-manual auto mode. First, the user can press the ND controller button to select items on the first level and enter the second level without releasing the NDH controller button, thus overlapping bi-manual actions. Secondly, if initiating hand is fixed (e.g. always start with NDH as in Figure ), to overcome the issue that the order of button pressing may be reversed in expert mode (pressing the DH controller first and then NDH controller in a very short period of time, for example, when the inverse should have been the case), a tolerance of 250ms was included in bi-manual VRLin. If, for example, the DH was mistakenly pressed and released less than 250ms ahead of NDH, bi-manual ordering can be adjusted to the correct sequence of actions, thus allowing enhanced temporal overlap of bi-manual actions.

## 4 PILOT STUDY

We conducted a series of pilot tests with an Oculus Rift and its two VR controllers to tune parameters. First, beginning with rotation angle, we use the controller handle angle to determine rotation. Controllers are held and rotated in the z-plane (see Figure 1) relative to the user (x is horizontal, y is vertical, z is depth). We chose this orientation because, if the participants in our pilot study pushed the front of the controller forward so the handles were more parallel with the

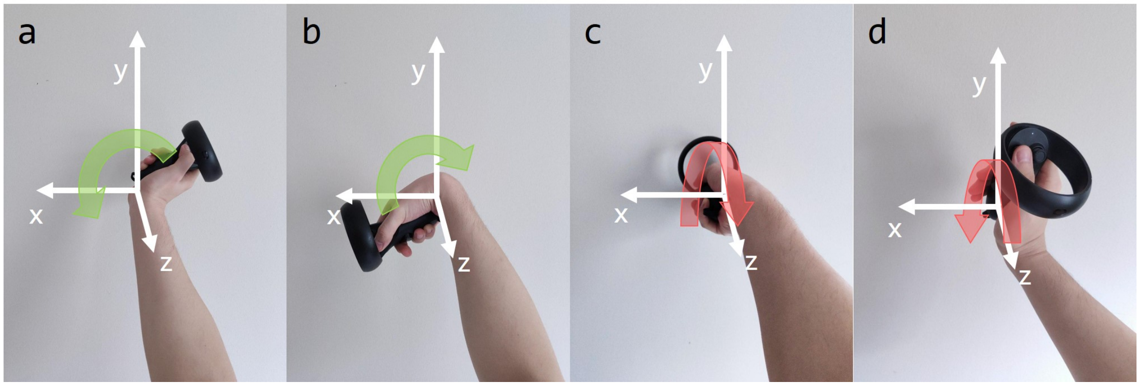



Fig. 1. To perform rotations in VRLin, users hold the controls vertically (handles at right angle to horizontal), as position that maximizes user comfort (a)(b). If users push the controller forward (handles parallel to the ground) (c) or pull back (d), participants experienced significantly more discomfort and their range of motion was limited. To guide users to the correct controller orientation, we used a halo which became increasingly transparent

z-axis (user-relative) or if they pulled back to position controllers more parallel the z-axis, it became much more difficult to rotation the wrist (i.e. observed rotational range was reduced) [1]. Figure 1 demonstrates this controller configuration. To guide the user to the correct orientation of the controller, we used an alpha-blended halo which became increasingly transparent as the user tilted the controllers out of the z-plane. As users gained experience with VRLin, and moved to auto mode, they had little problem maintaining an upright controller posture during command invocation.

Alongside mechanisms for guiding visual perception, we also explored the angular range of participant motion for menu invocation during our pilot tests. Participants were asked to move the controller through angles in the z-plane by rotating their wrists (pronation/supination of the wrist). Assuming that an angle of zero degrees is parallel to the ground in the z-plane and ninety degrees is vertical in the z-plane, we found an angular span of at least -25 degrees (controller head points downward to ground relative to handle at 25 degrees) to 150 degrees of movement for all of our participants with reasonable comfort of movement through this range. Note that participants could all exceed this range, but, because comfort was reduced, we used this range of motion to ensure that participants would not experience any undue discomfort with command activation. Any movement that exceeds this range is simply mapped onto the target located at the respective comfortable extremity of movement (movements beyond -25 degrees are mapped to -25 degrees and beyond 150 degrees are mapped back to 150 degrees respectively.

In the following two sections, we describe two studies of VRLin. The first study explores the number of targets participants can access using proprioceptive feedback. The second study presents a first-use evaluation of VRLin.

## 5 STUDY 1: DETERMINING MENU LAYOUT CARDINALITY

Initial work by Crossan et al. [11] in rotation-based input suggests target sizes of $9^o$ can be accurately acquired by the user, but it became clear in their experimental set-up (see Section 3 of [11]) that, while their goal was to eventually support eyes-free input, they leveraged visual feedback and allowed users to calibrate their input to determine what the

[1]Note that this aspect of wrist rotation – the collision between smooth pronation/supination movement and flexion/extension/Radial-Ulnar Deviation – is a known result from human motor control [11, 12, 29]. The primary cause of this difficulty is that flexion/extension/Radial-Ulnar Deviation both restricts forearm rotation at the elbow and interferes with axis-aligned wrist rotation along the wrist-hand axis.

tolerance was for eyes-engaged (non-proprioceptive) targeting. Visual monitoring resulted in movement times of 1.5s or more for target acquisition because their experimental design prioritized precision rather than eyes-free proprioception.

As we noted in our design of VRLin, our goal was to create a multi-level, sequential, rotation-based, in-air menu system that supports proprioceptive feedback and rehearsal-based learning. By design, VRLin's circular arrangement of sequential menus achieves many of these goals. However, supporting rehearsal-based learning through proprioceptive feedback requires some tuning. In this section, we describe a study that determines the number of items that can be reliably distinguished via proprioceptive feedback for both the dominant and non-dominant hands.

Recall that proprioception is an awareness of body position and movement [33]. This awareness of position and movement is important in motor planning because we do not need to attend to our body or to visually locate our hands, feet, and other body parts, to know where our limbs are positioned in the world relative to one another, to know how our bodies are moving, and to change body parts' relative direction of movement to achieve a desired configuration of our body. Specifically in the context of VRLin, we can know – without looking – the rotation of a controller we are holding simply via this proprioceptive awareness. We are also naturally aware of our head position relative to our feet, allowing us to define up and down relative to our bodies and relative to gravity. Controllers capture orientation using gravity sensors, and it is this awareness of relative positioning, i.e. this proprioception, and the pervasive presence of gravitational feedback captured by controllers, that we leverage in VRLin.

However, proprioceptive awareness of body position does not imply an unlimited ability to acquire an infinite number of targets via wrist rotation. Due to neurophysiological noise [33], i.e. imprecision in our ability to repeatedly place our limbs in an identical position, and due to imprecision in spatial estimation, there is a range of accuracy associated with any human motor control task. In the context of other rehearsal-based interfaces, such as marking menus [22], we are already aware of this constraint on target packing; it is used to limit the number of different targets per level of the marking menu, such that the menu is structured in hierarchical fashion. However, we are aware of no research that specifically explores the number of targets that can be accurately acquired in a rotational menu that leverages only proprioceptive feedback.

Given the above observations, the goal of this study is to find an appropriate menu layout that provides the most items with acceptable accuracy and speed for both the dominant (DH) and non-dominant (NDH) hands. Based on the approximate human ability in rotating their wrist from our pilot study, different numbers of items from 4 to 11 were tested within range of movement. Our goal was to maximize the number of items per menu level (cardinality) while preserving high accuracy (i.e. near perfect targeting ability using only proprioceptive feedback).

### 5.1 Participants and apparatus

We recruited 6 participants aged between 22 and 26 from a local university. All of them were right-handed.

The study was conducted on an Oculus Rift S with the Unity 2019.3.8f1 game engine running on a high-end Windows 10 PC (3.6GHz Intel i7-6850K CPU,GeForce GTX 1080 GPU). Custom software was constructed to prompt the user to perform rotational targeting tasks. Figure 3 shows the configuration of the interface for one trial during our experiment.

### 5.2 Study Design

*5.2.1 Instructions.* Participants were first asked to read a consent form describing the study purpose and procedure. They were then fitted with a VR headset, were asked to imagine a plane on the controller parallel to their body and vertical to the ground. Participants were instructed to rotate the controllers on that plane with their wrist. As per our pilot study, a transparent halo was used to guide users to the correct controller orientation.

*5.2.2 Experiment.* The experiment was a six-condition (number of targets) by two-factor (visual versus proprioceptive feedback) within subject design where condition was number of targets displayed to the participant. Participants began with a warm-up/training block that had 13 target positions that they were asked to acquire in random order (Table 1). Visual feedback was provided during warm-up.

Fig. 2. The setup for Study 1. The participant is rotating the controller with his non-dominant hand to select items.

After the warm-up condition, participants completed the six experimental conditions in an order determined via a Latin Square. There were four blocks in each condition and all items were selected once, in random order in each block. The items were were labeled with the angle from the target angle to horizontal as shown in Figure3. A text prompt indicated which item to select. The first two blocks of each condition were a with-visual-feedback task; in these blocks, the controllers and items including names and bars were visible. The second two blocks included no-visual-feedback of controller angle or target location; only the text prompt was shown in the VR scene. The conditions are listed in Table 1. Half of the participants started all of the task with the DH and the rest started with the NDH. Initial handedness had no impact on results.

In summary, 2 handedness * 4 blocks (2 with-visual-feedback + 2 no-visual-feedback) * (4+5+6+7+9+11) positions * 9 participants = 3024 data points were collected.

*5.2.3 Measures.* As this was a formative study to measure design factors for VRLin, we did not have formal hypotheses. We captured accuracy (angular error rate using most proximal angle to selected target) and task completion time from prompt to selection of angle.

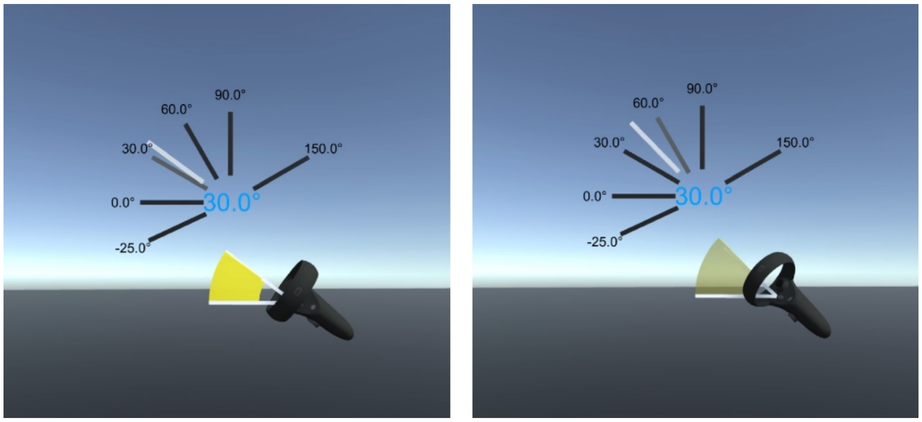



Fig. 3. Experiment platform design for study 1. Participants need to rotate the controller to select the prompt item (blue text, 30.0° in this case). The yellow halo visualizes the current angle and deviation from the required plane by changing the transparency of the arc.

Table 1. Conditions of Study 1

| condition | angles |
|---|---|
| warm-up | -25, -12.5, 0, 15, 22.5, 30, 45, 60, 67.5, 75, 90, 120, 150 |
| 4-item | -25, 0, 90, 150 |
| 5-item | -25, 0, 45, 90, 150 |
| 6-item | -25, 0, 30, 60, 90, 150 |
| 7-item | -25, 0, 22.5, 45, 67.5, 90, 150 |
| 9-item | -25, 0, 15, 30, 45, 60, 75, 90, 150 |
| 11-item | -25, -12.5, 0, 15, 30, 45, 60, 75, 90, 120, 150 |

### 5.3 Results

We removed 54 outlier data points (1.8%) due to long task completion times; long task completion times were typically a result of tracking or display issues. We used a repeated measures two-way ANOVA with "handedness" and "menu layout" as independent variable to highlight significant differences between groups.

The primary goal of this experiment is to tune our menu layout for proprioceptive feedback. Given this, we are particularly interested in eyes-free accuracy, where participants rely on proprioception to determine target location. If participants can accurately leverage proprioceptive feedback, this implies an ability to target without relying on a visual depiction of menu. As a result, we focus primarily on the final two blocks for each condition for each hand in our formative experiment, i.e. the blocks with no visual feedback.

### 5.4 Input Accuracy

Figure 4 shows user performance for input accuracy for no-visual-feedback task. A repeated measures ANOVA showed significant effect of both handedness ($F_{1,8} = 36.47$,p<.01) and menu layout ($F_{5,40} = 72.5$, p<.001). There was no significant interaction effect. Handedness (dominant vs non-dominant, had no effect on accuracy. Post hoc pairwise comparisons revealed significant difference between less than 6 items versus more than 7 items layouts (p<.001) but no difference between 4, 5, or 6 items layouts.

Examining, specifically, rotation angle, Figure 5 shows detailed input accuracy for different rotation angles, handedness ignored as not significant. With all menu layouts under 6 items, the participants could acquire the target accurately at any angle. With menu layout with 11 items, although overall accuracy was low, participants still could acquire targets at 90, 120 and 150 degrees with high accuracy.

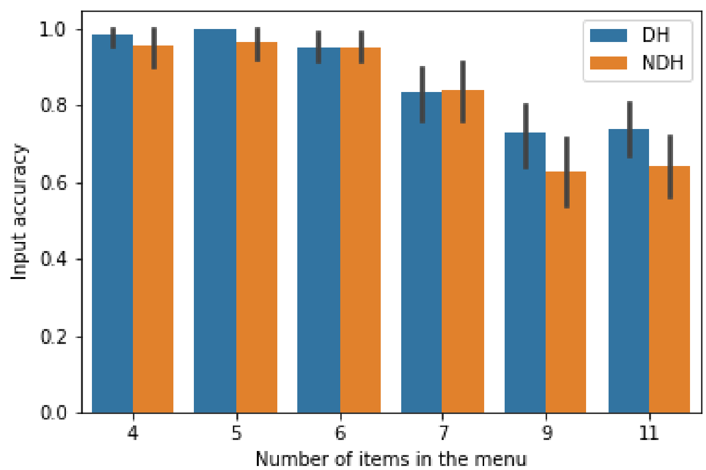



Fig. 4. Input accuracy of different number of items in the menu for no-visual-feedback task. Error bars are 95% CI.

### 5.5 Task Completion Time

Figure 6 shows task completion time for each menu layout. A repeated measures ANOVA showed significant effect of menu layout ($F_{5,40}$ = 24.8, p<.001) but no effect of handedness was found. A post hoc pairwise test revealed significant difference between 5 items layout and 7 or more items layouts (p<0.05), as well as between 6 items layout and 9 or 11 items layouts (p<0.05). We note that no visual feedback did increase task completion time. For 4 to 6 item layouts, this increase was approximately 200ms.

### 5.6 Synthesis

To support proprioceptive targeting, our goal is for users to be able to target reliably without visual feedback or, even in the presence of visual feedback, without needing to visually attend to the target. This goal is consistent with other rehearsal-based interfaces such as marking menus and FastTap. Analysis of our data indicates that users can accurately target up to six items, with items placed at -20 degrees, 0 degrees, 30 degrees, 60 degrees, 90 degrees, and 150 degrees. On further analysis of our data for 11 items, we also noted highly accurate discrimination between targets at 90 degrees, 120 degrees, and 150 degrees. However, examining Figure 5, the same is clearly not true for angle less than 0 degrees; the introduction of a target at -12.5 degrees significantly impacted accuracy at -25 degrees and 0 degrees. We conducted

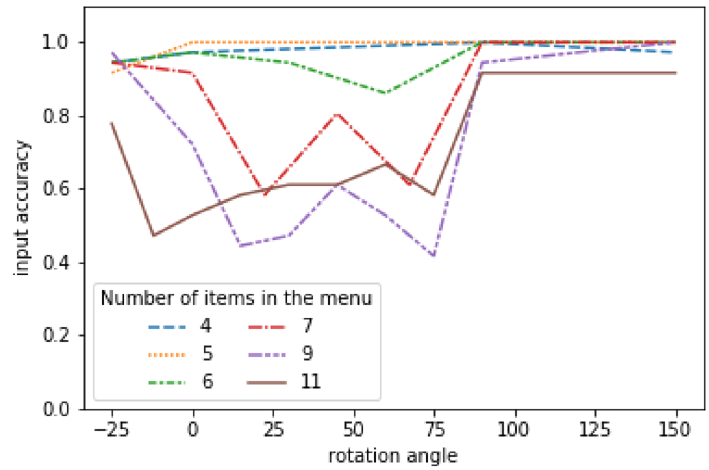



Fig. 5. Input accuracy of different rotation angle for no-visual-feedback task.

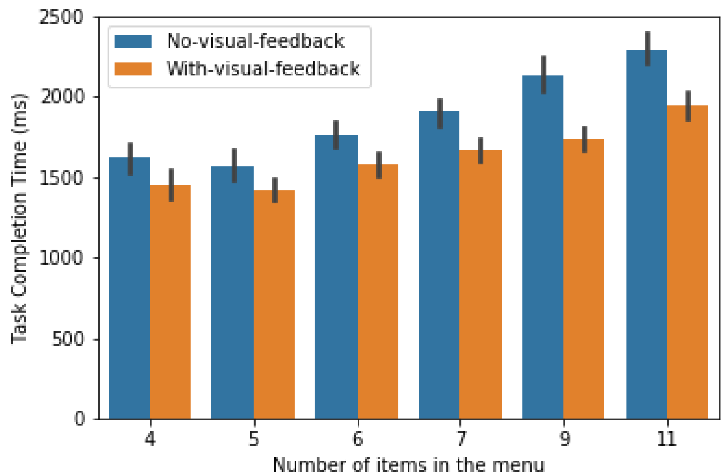



Fig. 6. Task completion time of different number of items in the menu. Error bars are 95% CI.

an additional pilot study using seven items (targets centred at -20, 0, 30, 60, 90, 120, and 150 degrees) regardless of handedness, and observed consistently high accuracy. This yielded an initial design recommendation of seven targets per level for both dominant and non-dominant hands.

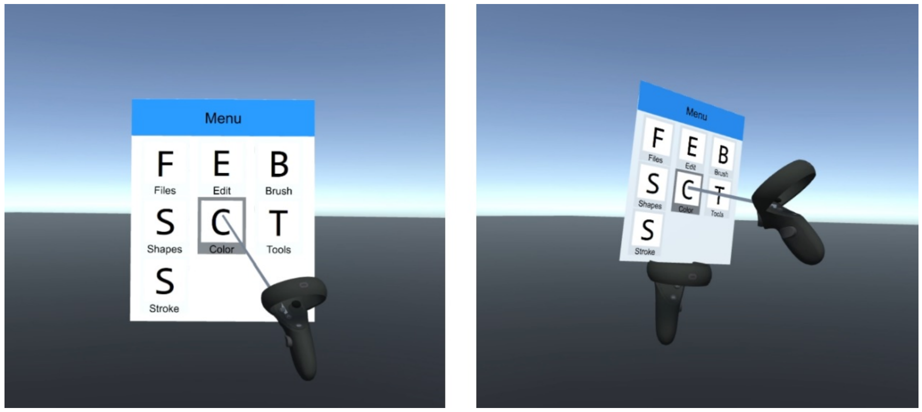



Fig. 7. The control interfaces for Study 2. Left is the wall placement palette menu and right is the hand placement palette menu. VRLin conditions are shown in Figure

## 6 STUDY 2

The previous study shows that users can select 7 items both quickly and accurately for both hands. To explore the performance of VRlin menu in comparison to other VR-based command selection mechanisms, we conducted a second study.

In this section we report the results of a design review assessing our technique against two competing, contemporary command selection techniques that have been used in VR [2]. Our design review captures both quantitative and qualitative data but, because of the limited sample size, descriptive values and qualitative assessments are stressed in this section.

### 6.1 Study Design

*6.1.1 Participants and Apparatus.* Six participants aged between 19 to 40 participated in our study. All were owners of Oculus headsets. Two participants (P1 and P4) had significant VR experience. Other participants, while familiar with VR, had less experience.

This study was conducted remotely. The study was run on the Oculus Rift S with the Unity game engine run on participants’ personal computers with Windows 10 operating system in their home VR environment.

*6.1.2 Application.* For this study, we designed a VR painting application and constructed four variants of the painting application’s interface: There were two traditional palette menu layouts as shown in Figure 7. They shared similar layouts. Figure 7 left is a wall placement palette menu which is popular for current VR applications [2]. Figure 7 right is a hand placement palette menu which is similar to VR drawing or modeling applications[3, 4]. Uni-manual and a bi-manual VRLin variants were also implemented to contrast the bi-manual command selection strategy. The uni-manual variant incorporates one-hand-sequential selection as follows: users press and release with the DH to select the first level item, then do the same to select the second level item. Users do not alternate hands during command selection, as in Figure . We label these command conditions W-Pal, H-Pal U-VRLin, VRLin for Wall-Palette, Hand-Palette, Unimanual-VRLin and Bimanual-VRLin respectively.

[2]While our initial plan was for a quantitative study, our institution suspended all human subjects experiments that required direct contact with participants. As a result, we conducted a remote study with owners of personal VR set-ups. We characterize this as a design review due to the nature of the participant group, the lack of controlled environment, and the non-consistent computer hardware used.

To encourage rehearsal-based learning, we incorporated a delay of 250ms in menu appearance in both VRLin conditions. We acknowledge that the presence or absence of delay and the length of delay is debatable [17, 26] in desktop interfaces. However, given the processing required for 3D scene rendering and the immersion that we attempt to preserve in virtual environments, we incorporate a common value for delayed menu drawn from the Maya desktop application (250ms) as it may speed UI rendering and may reduce visual disruption of the 3D scene. Delay was only present in the VRLin and U-VRLin conditions. In other conditions, because visual targeting was essential, no delay was included in menu appearance.

Our hierarchical command structure leveraged the 7-item limit identified in the first experiment. There were 7 items on the first level and 7 * 7 items on the second level with icons and names. To facilitate transitions to expert use (i.e. non-visual feedback), only 4 items were random selected out of 7 * 7 items in total for participants to choose during a prompted study.

The system logged all user actions, including time from prompt to selection and selection errors. This data was stored in a file within the application directory.

*6.1.3 Study Procedure.* The study was conducted using video conferencing software of the participants’ choice. Participants were recruited via email and provided with consent documents prior to participation. At a mutually convenient time, participants joined a video call, were greeted, and the purpose of the study was explained to them. Participants were informed that participation in the study indicated implied consent for their data to be used in publications related to this work, that they could cease participation at any time, and that no data would be transmitted from their computer automatically, i.e. that experimental data was stored locally and that, if they agreed, then they could copy and transmit to us their logged data after the experiment was complete.

After participants were comfortable with their rights, the experiment started. Participants obtained an executable version of the software, unpacked it into a specified directory, and tested its functionality. Participants were then given instructions on each menu technique before they began an individual condition. They were asked to select 5 items as practice to develop experience with the technique.

The experiment was a four-condition within-subjects experiment with command condition (W-Pal, H-Pal, U-VRLin, VRLin) as the independent variable. Order of presentation of condition was counter-balanced across participants using a perfect Latin Square.

Within each condition, participants performed 24 blocks. Within each block, participants selected four different menu items in random order (the same four items, but ordered differently/randomly for each block), i.e. four trials per block. For each trial (selection), a stimulus icon and text appeared in the air in front of the participant. The participant would select the corresponding item from the menu, and then draw a stroke through a ball positioned in front of the participant to complete the trial (see accompanying video). If the participant began and completed their rotation in less than 250ms, no menu was shown; otherwise a menu appeared.

We note that, during interaction with H-Pal and W-Pal, the hand and wall palettes remained visible and the pointer could remain on the palette. We acknowledge that this unfairly advantages both H-Pal and W-Pal. However, it is also the case that forcing a user to target elsewhere in the environment and then re-invoke, re-visually acquire, and perform distant targeting could also be designed to unfairly disadvantage H-Pal and W-Pal. As a result, our experiment does not test specific additional advantages of rotational menu access (e.g. the lack of need to visually locate the menu in space before targeting, the lack of need to perform Fitts’s style targeting) nor additional advantages of proprioceptive/eyes-free command access (i.e. the lack of need to display a menu in space). These are *a priori* advantages of VRLin, which we felt no need to assess in a design review.

After the experiment, participants were asked to copy and forward data from the application log file to us via the chat window in video conferencing software. Participants then participated in a semi-structured interview garnering their opinions and comments on the four command layouts.

### 6.2 Results

In this section, we present both quantitative and qualitative assessments from participants.

*6.2.1 Quantitative (Log) Data.* Figure 8 shows the time from prompt to selection for participants across conditions. Recall that, because participants could remain on the command selection palette, i.e. they did not need to target the command palettes, simply target within the command palettes, these times should be considered *better than best case* estimates for H-Pal and W-Pal's temporal performance. We also note that we report total time, including the delay time for menu invocation with both bi-manual and uni-manual VRLin techniques. We recognize that this decision (to penalize VRLin's menu mode, or "unpracticed" performance) is controversial; as noted by Lewis [26], for fair comparison of unpracticed performance during initial trials, it would be more appropriate to subtract this delay. However, it is relatively easy to perform this subtraction analytically and note that, based on descriptive statistics and *in the absence of delay* unimanual VRLin (U-VRLin) initially outperforms other techniques (when averaged across participants) and bi-manual VRLin performs on-par with wall palette (W-Pal) and under-performs hand palette (H-Pal). Furthermore, based upon Figure 8, H-Pal achieves optimal performance earlier, while other techniques appear to continue to improve in terms of completion time throughout the study.

Looking specifically at Figure 8, all times including delay for VRLin variants, we note uni-manual and bi-manual VRLin appear to perform on-par with wall palettes. While this neglects pointing and visual time to acquire the relative position of targets within the wall palette, we would argue that it shows that, once within the command invocation system, VRLin's variants appear approximately on-par with Wall-based Palette command invocation, i.e. with Palettes displayed in physical space on a plane in front of the user. Hand palette outperforms other techniques specifically for command access (neglecting time required to point to the command palette); however, hand palette techniques would require additional Fitts's Law targeting times because they are placed on the non-dominant hand, i.e. they cannot be conveniently placed directly in front of where the controller is pointing, so the user would need to move a ray to acquire the hand palette. In all cases, differences are small once the palette/command invocation mechanism is acquired. Furthermore, because VRLin does not require spatial targeting (menus can simply appear in front of controllers) and because it can be performed eyes-free (without disruption to the virtual world), there are significant, unassessed advantages of VRLin.

Considering eyes-free input, i.e. the elimination of visual disruption, one question we posed was whether VRLin had achieved fully practiced performance during the study, or whether participants were still optimizing their behaviors. We note that participants had approximately 10 minutes to interact with the VRLin conditions, so it would be unsurprising if they did not receive practiced performance, i.e. consistent use of the autonomous mode (Auto-Mode).

Our log files confirm that, while participants begin to use Auto-Mode on the first menu level, on the second menu level the use of Auto-Mode remains relatively infrequent (see Figure 9). Participants used Auto-Mode approximately 40% of the time in the first level, and less than 10% of the time in the second level in the uni-manual variant, and, in VRLin's bi-manual variant, the use of Auto-Mode was approximately half that rate. As a result, participants have the

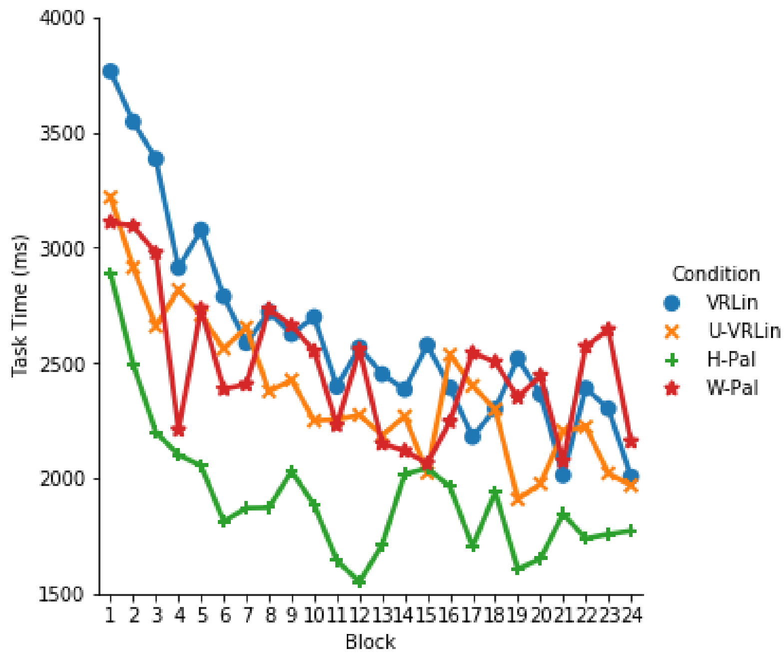
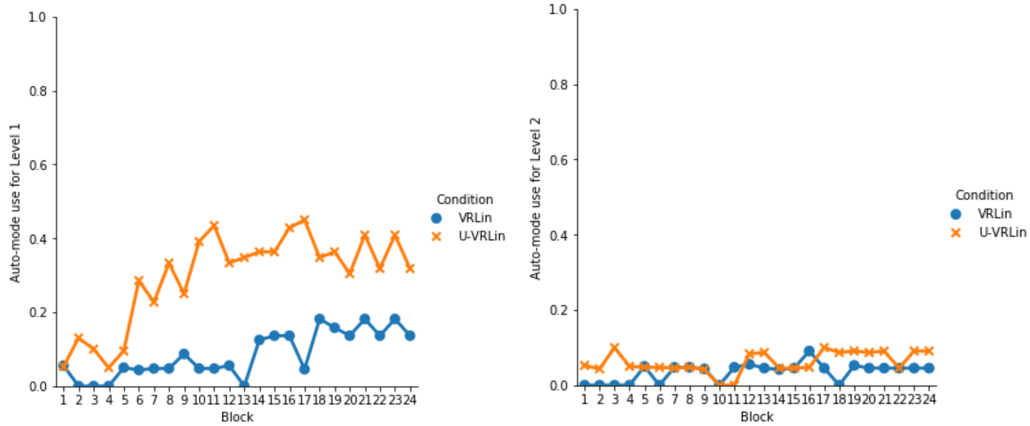



Fig. 8. Task Time per block (from prompt to select) for each technique.

potential to continue to improve in their temporal performance compared to other techniques. While a theoretical performance gain of 375ms and 450ms is possible, we note that it is rare for users to achieve practiced performance in every command item. However, because command use typically follows a zipfian distribution [10], it is highly likely that participants can achieve improved overall performance, and, in particular, improved performance on frequently accessed commands. The question then becomes whether a small cost in command access (versus H-Pal) is worth the overall benefit of eyes-free command access for more practiced (i.e. presumably more frequently used) commands.

Finally, we administered the NASA TLX [1] to participants and present component scores (20-point scale) in Figure 10. While, on average, visual inspection reveals that, qualitatively, VRLin performance is on-par with wall palette techniques and underperforms hand palette, differences are relatively small between uni-manual VRLin and other techniques. We note the bi-manual VRLin underperformed all other techniques, including uni-manual VRLin, across all components, undoubtedly due to the additional cognitive cost associated with bi-manual coordination.

*6.2.2 Qualitative Assessments.* Over our six participants, in terms of preference, three preferred VRLin. The other three preferred one or both variants of the palette menu. However, one additional participant, P6, noted that, if given additional time to learn, this participant's preference would switch to the rotational menu, VRLin. The wall palette, being more familiar, is easier to interact with initially. For the three participants who preferred VRLin, two perferred the uni-manual variant. Overall, contrasting uni-manual and bi-manual variants of VRLin, participants were equally split. In particular, participants found they could achieve higher performance during the study with the uni-manual variant, but there was also an impression that, with additional practice, the bi-manual variant had potential to significantly increase in terms of performance.




(a) (b)

Fig. 9. Auto-Mode Use for first (a) and second (b) menu levels.

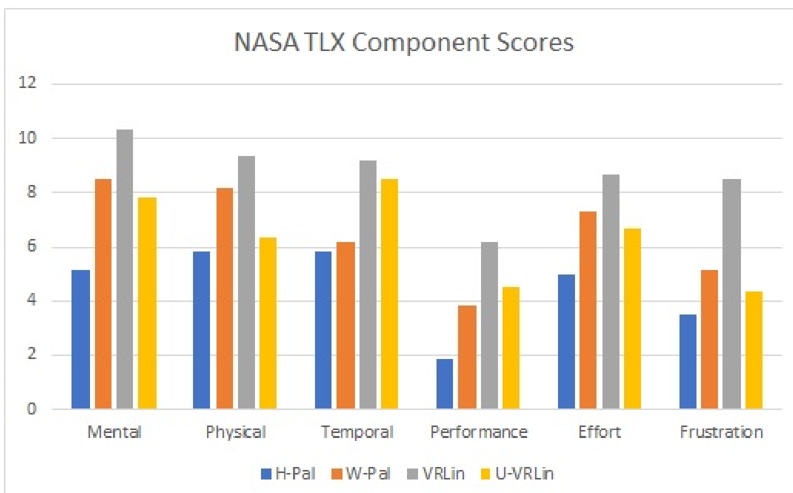



Fig. 10. NASA TLX Component Scores for each technique.

In terms of number of items, all participants found that the number was appropriate for accurate targeting, though one participant did note that having one or two fewer items per level might further speed performance for that level, but at the cost of deeper menus if a large command set is needed. The trade-off between breadth and depth in terms of time and learning is one that could be explored with a significantly larger-scale, in laboratory experimental study.

Finally, one participant noted that many controllers are capable of haptic feedback, and this could be leveraged to significantly improve the speed and accuracy of rotational menus, particularly for eyes-free input. While we considered leveraging haptic feedback, in this study we chose not to as we felt that haptics would introduce an advantage to VRLin, an advantage that could not be replicated in palette menus.

## 7 DISCUSSION AND CONCLUSION

Our first study in this paper explore users' ability to target using eyes-free wrist rotation via a two-phase study where participants learn with feedback and then attempt eyes-free the careful selection of angular targets using wrist rotation. Our results argue for a seven target design for both dominant and non-dominant hands with targets centred at -20, 0, 30, 60, 90, 120, and 150 degrees. This represents a follow-on contribution to the work of Crossan et al. [11] on angular selection of targets using wrist rotation.

Our second study shows that VRLin's performance is on par with Wall palette and Hand Palette for command selection, especially in menu mode during unpracticed use. However, it is important to note that these performance values are also an *underestimate* of the performance of VRLin. To understand why our results significantly understate the performance of VRLin, note that we specifically neglect the time required to move to and then target the Hand Palette, and we also eliminate the need to visually acquire the wall palette. As well, wall palettes must either positioned at the location of the current controller ray (blocking content behind it) or to the side but then targeted with the controller. These are additional costs of palette-based selection that are eliminated by VRLin. Given that these advantages to VRLin exist vacuously, we felt a better focus would be on measuring the selection complexity of VRLin compared to hand and wall palette once these menus were acquired.

In summary, this paper describes VRLin, a sequential, rotational menu for eyes-free, proprioceptive HMD command access. Through a pilot study and an initial design study, we find that participants can accurately target up to seven items using only proprioception with both dominant and non-dominant hands, and provide guidance for angular placement of these items. Through a second design review study, we demonstrate that VRLin represents a promising mechanism to support eyes-free, proprioceptive command selection in immersive virtual environments.

## REFERENCES

[1] NASA-TLX Questionnaire. https://humansystems.arc.nasa.gov/groups/TLX/. (????).
[2] 2014. SteamVR on Steam. (2014). https://store.steampowered.com/app/250820/SteamVR/
[3] 2016. Tilt Brush. (2016). https://www.tiltbrush.com/
[4] 2017. Blocks - Create 3D models in VR. (2017). https://arvr.google.com/blocks/
[5] Takumi Azai, Shuhei Ogawa, Mai Otsuki, Fumihisa Shibata, and Asako Kimura. 2017. Selection and Manipulation Methods for a Menu Widget on the Human Forearm. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 357–360. DOI:http://dx.doi.org/10.1145/3027063.3052959
[6] Gilles Bailly, Eric Lecolinet, and Yves Guiard. 2010. Finger-Count Radial-Stroke Shortcuts: 2 Techniques for Augmenting Linear Menus on Multi-Touch Surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 591–594. DOI:http://dx.doi.org/10.1145/1753326.1753414
[7] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2008. Flower Menus: A New Type of Marking Menu with Large Menu Breadth, within Groups and Efficient Expert Mode Memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. Association for Computing Machinery, New York, NY, USA, 15–22. DOI:http://dx.doi.org/10.1145/1385569.1385575
[8] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. 2011. Comparing Free Hand Menu Techniques for Distant Displays Using Linear, Marking and Finger-Count Menus. 248–262. DOI:http://dx.doi.org/10.1007/978-3-642-23771-3_19
[9] Mark Billinghurst, Sisinio Baldis, Lydia Matheson, and Mark Philips. 1997. 3D palette: a virtual reality content creation tool. In *Proceedings of the ACM symposium on Virtual reality software and technology*. 155–156.
[10] Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A Predictive Model of Menu Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 627–636. DOI:http://dx.doi.org/10.1145/1240624.1240723
[11] Andrew Crossan, John Williamson, Stephen Brewster, and Rod Murray-Smith. 2008. Wrist Rotation for Interaction in Mobile Contexts. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. Association for Computing Machinery, New York, NY, USA, 435–438. DOI:http://dx.doi.org/10.1145/1409240.1409307
[12] E. R. F. W. Crossman and P. J. Goodeve. 1983. Feedback Control of Hand-Movement and Fitts' Law. *The Quarterly Journal of Experimental Psychology Section A* 35, 2 (1983), 251–278. DOI:http://dx.doi.org/10.1080/14640748308402133

[13] Raimund Dachselt and Anett Hübner. 2007. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics* 31, 1 (2007), 53–65.

[14] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen. 2013. Extended Pie Menus for Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (2013), 644–651.

[15] Yves Guiard. 1988. The kinematic chain as a model for human asymmetrical bimanual cooperation. In *Advances in Psychology*. Vol. 55. Elsevier, 205–228.

[16] Carl Gutwin, Andy Cockburn, Joey Scarr, Sylvain Malacria, and Scott C. Olson. 2014. Faster Command Selection on Tablets with FastTap. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2617–2626. DOI: http://dx.doi.org/10.1145/2556288.2557136

[17] Jay Henderson, Sylvain Malacria, Mathieu Nancel, and Edward Lank. 2020. Investigating the Necessity of Delay in Marking Menu Invocation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI'20)*. 13.

[18] Jay Henderson, Sachi Mizobuchi, Wei Li, and Edward Lank. 2019. Exploring Cross-Modal Training via Touch to Learn a Mid-Air Marking Menu Gesture Set. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 8, 9 pages. DOI: http://dx.doi.org/10.1145/3338286.3340119

[19] Ankit Kamal, Yang Li, and Edward Lank. 2014. Teaching Motion Gestures via Recognizer Feedback. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. Association for Computing Machinery, New York, NY, USA, 73–82. DOI: http://dx.doi.org/10.1145/2557500.2557521

[20] Keiko Katsuragawa, Krzysztof Pietroszek, James R Wallace, and Edward Lank. 2016. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*. 128–135.

[21] Konstantin Klamka, Patrick Reipschläger, and Raimund Dachselt. 2019. *CHARM: Cord-Based Haptic Augmented Reality Manipulation*. 96–114. DOI: http://dx.doi.org/10.1007/978-3-030-21607-8_8

[22] Gordon Paul Kurtenbach. 1993. *The Design and Evaluation of Marking Menus*. Ph.D. Dissertation. CAN.

[23] Benjamin Lafreniere, Carl Gutwin, Andy Cockburn, and Tovi Grossman. 2016. Faster Command Selection on Touchscreen Watches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4663–4674. DOI: http://dx.doi.org/10.1145/2858036.2858166

[24] Celine Latulipe. 2004. Symmetric interaction in the user interface. *Companion Proceedings of UIST* (2004), 63–66.

[25] Julien-Charles Lévesque, Denis Laurendeau, and Marielle Mokhtari. 2013. An asymmetric bimanual gestural interface for immersive virtual environments. In *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 192–201.

[26] Blaine Lewis. 2019. *Longer Delays in Rehearsal-based Interfaces Increase Expert Use*. Master's thesis. University of Waterloo.

[27] Sylvain Malacria, Gilles Bailly, Joel Harrison, Andy Cockburn, and Carl Gutwin. 2013. Promoting hotkey use through rehearsal with exposehk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 573–582.

[28] P. Monteiro, H. Coelho, G. Gonçalves, M. Melo, and M. Bessa. 2019. Comparison of Radial and Panel Menus in Virtual Reality. *IEEE Access* 7 (2019), 116370–116379.

[29] Tony Morelli and Eelke Folmer. 2012. Twuist: A discrete tactile-proprioceptive display for eye and ear free output on mobile devices. In *2012 IEEE Haptics Symposium (HAPTICS)*. IEEE, 443–450.

[30] Krzysztof Pietroszek and Edward Lank. 2012. Clicking Blindly: Using Spatial Correspondence to Select Targets in Multi-Device Environments. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '12)*. Association for Computing Machinery, New York, NY, USA, 331–334. DOI: http://dx.doi.org/10.1145/2371574.2371625

[31] Krzysztof Pietroszek, Liudmila Tahai, James R Wallace, and Edward Lank. 2017. Watchcasting: Freehand 3D interaction with off-the-shelf smartwatch. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 172–175.

[32] Majid Pourmemar and Charalambos Poullis. 2019. Visualizing and Interacting with Hierarchical Menus in Immersive Augmented Reality. In *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 30, 9 pages. DOI: http://dx.doi.org/10.1145/3359997.3365693

[33] David A Rosenbaum. 2009. *Human motor control*. Academic press.

[34] Jaime Ruiz, Yang Li, and Edward Lank. 2011. User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 197–206.

[35] Md. Sami Uddin, Carl Gutwin, and Benjamin Lafreniere. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5836–5848. DOI: http://dx.doi.org/10.1145/2858036.2858211

[36] Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1083–1092.

[37] Yukang Yan, Chun Yu, Xiaojuan Ma, Shuai Huang, Hasan Iqbal, and Yuanchun Shi. 2018. Eyes-Free Target Acquisition in Interaction Space around the Body for Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 42, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3173616

[38] Shengdong Zhao, Maneesh Agrawala, and Ken Hinckley. 2006. Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-Stroke Marking Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1077–1086. DOI: http://dx.doi.org/10.1145/1124772.1124933

[39] Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. Association for Computing Machinery, New York, NY, USA, 33–42. DOI: http://dx.doi.org/10.1145/1029632.1029639

[40] Suwen Zhu, Jingjie Zheng, Shumin Zhai, and Xiaojun Bi. 2019. I'sFree: Eyes-Free Gesture Typing via a Touch-Enabled Remote Control. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 448, 12 pages. DOI: http://dx.doi.org/10.1145/3290605.3300678