Drum Menu: Bimanual Controller Command Access Techniques in Virtual Reality

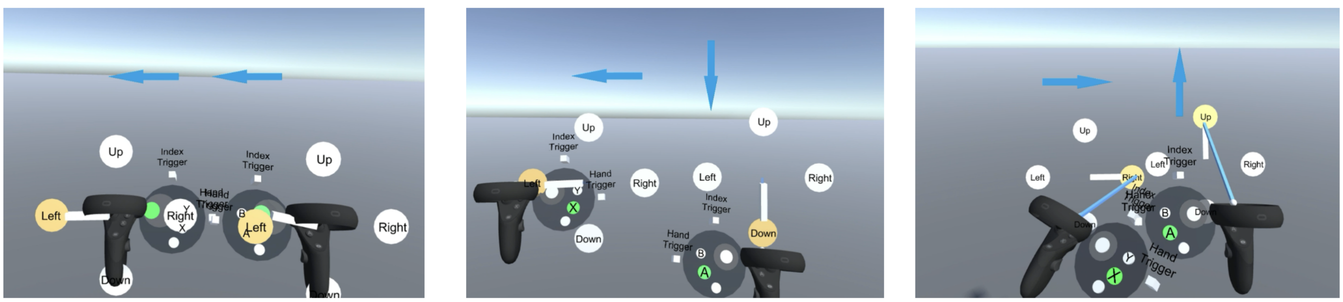
Futian Zhang
Cheriton School of Computer Science, University of Waterloo
Canada
futian.zhang@uwaterloo.ca

Paul Kokhanov
Engineering, University of Waterloo
Canada
pkokhanov@uwaterloo.ca

Edward Lank
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
lank@uwaterloo.ca

Keiko Katsuragawa
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
National Research Council Canada
Waterloo, ON, Canada
kkatsuragawa@uwaterloo.ca

Jian Zhao
Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
jianzhao@uwaterloo.ca



a) b) c)

Figure 1: Drum Menu is a bimanual rapid command access technique in virtual reality (VR) with three variants derived from 2-level pie menu. a) Bimanual Joystick Drum Menu. Users begin with rotating the joystick on non-dominant controller to different angles to select the first-level item, then the joystick on the dominant controller to select the second-level item. They could select two levels sequentially or simultaneously, which also applies to the rest Drum Menus. b) Bimanual Stroke. Similar to pen-based marking menu, users can press a button to draw a stroke pointing to the item in the vertical plane. Users need to access two levels with their non-dominant controller and dominant controller. They can release the button on the dominant controller to confirm selection. c) Bimanual Rotation Drum Menu. The menu is always positioned in front of the controllers. Users can rotate the controller upward a bit to point to the upper item, or leftward to select the left item, same for the rests. They need to access two levels with their non-dominant controller and dominant controller. There will be a short delay between the button being pressed and menu being visible.

ABSTRACT

Current Virtual Reality (VR) Head-Mounted Displays (HMDs) offer limited shortcuts for rapid command access, which often requires users to navigate menus through precise visual targeting at multiple depths. This process can be slow and distracting, particularly during immersive gaming or productivity tasks. While marking menus have shown effectiveness as a shortcut command access mechanism, their performance in VR has not been adequately studied. Moreover, their potential integration with 6-degree-of-freedom (6-DoF) controllers and 2-DoF joysticks in VR environments remains largely unexplored. In this paper, we introduce the Drum Menu, a bimanual shortcut command access technique derived from idea of traditional pie menus, featuring three input methods, designed for 4-item and 8-item layouts specifically for VR controller command access. Users can select commands by rotating the joystick, drawing a stroke, or pointing in different directions. Bimanual input enables simultaneous access to two menu levels. A controlled user study reveals that drum menus are faster than the unimanual versions for the 4-item layout. Additionally, users prefer the bimanual joystick drum menu with the 4-item layout given its short task time, low error rate and low physical movement. For the 8-item layout,

stroke drum menus are found to be less error-prone for expert users compared to the other techniques.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**.

KEYWORDS

Virtual reality, controllers, marking memus, shortcuts, interaction techniques, controlled study.

ACM Reference Format:
Futian Zhang, Paul Kokhanov, Edward Lank, Keiko Katsuragawa, and Jian Zhao. . Drum Menu: Bimanual Controller Command Access Techniques in Virtual Reality. In *Proceedings of* . ACM, New York, NY, USA, 11 pages.

1 INTRODUCTION

Alongside the input of data or text [38], one frequent target of research innovation is command selection [37]. Besides menus and buttons [5], researchers have explored how best to access command shortcuts [6, 28], how to design command gestures [35, 37], and how to leverage rehearsal [16, 22] and cross-modal learning [17, 41] to support rich, rapid command selection for 2D displays.

Recently, virtual reality (VR) experiences created by Head-Mounted Displays (HMDs) have been becoming more and more popular in various applications including gaming and productivity tasks. However, it presents unique challenges for command selection. Navigating a menu and inputting shortcuts are two common ways that users utilize to select commands on computing devices. Currently, most VR applications use menus to support command selection, which highly requires visual engagement and ray pointing [14]. They could be friendly and familiar to novices but become slow and disruptive for expert users and advanced use cases [6]. Users have to spend time on finding the item and match the item with the expected action in their mind, and a visible menu will occlude to the main content[22]. On the other hand, due to a lack of buttons on controllers, compared with keyboards, shortcuts are rarely available for VR HMD users. With the emergence of productivity tasks on VR devices [1, 12], there is a demand for developing rapid command selection mechanisms.

Marking menu is a radial menu technique allowing users to select commands by drawing a quick stroke pointing to the item. It supports both novice (GUI) and expert (gestures) users with a mechanism of transiting from novice to expertise [7]. Most of the previous marking menu techniques are developed for 2D interfaces like desktops or tablets, with single 2D input devices like mouse and touch screen. Mid-air marking menus were also explored [14, 26, 34], but only unimanual input was discovered, and to the best of our knowledge, expert mode performance has not been evaluated. Furthermore, all of the VR marking menu techniques were only mapping 3 Degrees-of-Freedom (DoF) positional or 3 DoF directional information to 2D space [14], and compared among them. However, most of the current VR devices come with two controllers with 3-DoF positional and 3-DoF directional information plus additional 2D touchpads or joysticks. Thus, two possible avenues remain the potential to increase efficiency: 1) fully investigate the untouched input information including 3-DoF positional, directions, and 2D touchpad/joysticks input, and 2) leverage asymmetric bimanual input to facilitate multi-level menu access.

To fill in the gaps exploring these avenues, in this paper, we introduce Drum Menu (Figure 1), a bimanual rapid command access technique derived from radial menus, tailored for VR controllers. To address the two possible avenues mentioned above, Drum Menu explores: 1) three distinct ways to map 3 DoF positional, 3 DoF directional, and 2D joystick input to a 2D marking menu layout (i.e., Stroke Drum Menu, Pointing Drum Menu, and Joystick Drum Menu), and 2) integrated with bimanual interaction. All of the Drum Menus are using two-level radial menu layout which is similar to marking menus. Joystick Drum (Figure 1a, Figure 1b) Menu will use the joystick angle to select the menu items. Stroke Drum Menu (Figure 1c, Figure 1d) is similar to the traditional marking menu, which maps the 3D position trace to a 2D stroke on a plane parallel to the VR HMD. Pointing Drum Menu (Figure 1e, Figure 1f)uses the pitch and yaw dimension of the controller, mapping the tilt angle of the controller to the pointing direction to select menu items. Bimanual interaction allows users to access two levels at the same time. Two menus would be associated with both hands. The non-dominant hand controls the first-level menu and the dominant hand controls the second-level menu. The user selects the first-level menu with the non-dominant hand, and the second-level menu will be changed according to the first-level menu selection. Then, the user confirms the command in the second level with the dominant hand. When users are familiar with the command position for both hands, they can access the command from two levels at the same time.

We conducted a controlled experiment to evaluate the performance of each Drum Menu under a different number of items. To evaluate the benefits of bimanual interaction, we also developed unimanual variants in the user study, which would require users to select the two levels sequentially just like marking menus. We designed a command access task with a 4-item menu layout and an 8-item layout and evaluated each technique based on the task completion time, and error rate for general performance. To better understand each technique, we also measured total hand distance and total hand rotation, and asymmetric time which is the time difference between selecting the first level and second level for bimanual variants. We found that the 4-item layout is a better layout in terms of accuracy and speed, however, the 8-item layout performs much worse than the 4-item layout. Among those 4-Item layouts, the Joystick Drum Menu represents an effective mechanism to support shortcut command selection in immersive virtual environments, followed by the Stoke Drum Menu. For the 8-item layout, Stroke has the lowest error rate compared with others.

In summary, the contributions of this paper include:

- A bimanual rapid command access technique, called Drum Menu, in virtual reality (VR) with three variants (Stroke Drum Menu, Pointing Drum Menu, and Joystick Drum Menu) derived from a 2-level pie menu.
- An in-depth analysis of results for participants' performance among three variants compared with the unimanual version, which further generates design implications.

2 BACKGROUND AND RELATED WORK

In this section, we first introduce the background of 2D marking menus, then describe relevant work in the areas of command selection in virtual reality.

2.1 2D Marking Menus

Marking menu is a radial menu interface that enables users to select commands by making quick gestures or marks. It supports both novice mode and expert mode. Novice mode requires visual search to locate the item then select with gesture or stroke, which can be time-consuming and disruptive. In expert mode, selection could be done instantly without visual search [4, 22, 39, 40]. In these marking menus, the user starts in expert mode without a visible menu, but can easily switch to novice mode by displaying a GUI menu. When the user places a stylus or pointer in the input space, a radial menu appears, allowing them to select an item by stroking in the corresponding direction (novice mode). As a result, it enables smooth transitions from novice to expert usage. Kin et al. developed a bimanual marking menu technique for touch screen, which reveals the bimanual variant is 10% 15% faster than the unimanual marking menu [19]. However, how to integrate bimanual interaction in 3D VR environment and the performance remains unknown. Shengdong et al. developed Simple Marking Menu that breaks multi-level gestures from one single "zig-zag" compound mark to several simple strokes. [40] It eliminated ambiguous Marks due to scale invariance and reduced space requirement. Inspired by the user interactions with 2D marking menus, we propose Drum Menu, designed for use in VR environments. More specifically, we adopted the Simple Marking Menu technique in the Stroke variant Drum menu due to the benefits it provided.

2.2 VR Command Selection

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. For example, tutorials on creating user interfaces using Unity leverage the UIHelper prefab to support ray cast 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 [10, 21, 36]. Specifically considering pie menus, the menu design adopted by Drum Menu, there are several variants of hierarchical pie menus for 3D virtual environments [14, 20, 31, 33]. Azai et al. introduced an on-body hierarchical menu in mixed reality [3], 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. Armstrong [27] also explored the menu around the non-dominant arm and how to optimize the layout of such arm-anchored UI. However, these on-body menus are not designed for shortcuts as they require visual engagement. Monteiro et al. explored the performance of radial and panel menus with two placements (fixed on the wall or 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 [29]. Santos et al. discovered that the selection time on radial menus is faster than that of linear menus with a ray cast interaction, where one uses a directional mark (as in marking menus [22]) to access menu items. Wentzel et al. found that in VR, for two-level menus, the marking menu is faster than other menus for a small number of items (8 in their case)[36]. StickyPie[2] is a marking menu technique that allows for scale-independent input by estimating the landing positions of saccades, enabling expert-level behavior. However, it requires additional eye-tracking hardware which is less popular and more expensive. While many such techniques exist, none specifically has explored the effectiveness of sequential bi-manual hierarchically arranged and access commands as we do in our studies.

2.3 Rotational Command Selection

Drum Menu design involves wrist rotation command selection in Rotation variants. Wrist rotation has been explored more generally in human motor control research, and it is known that rotational targeting rotational wrist positions follow Fitts's Law (confirmed initially by [9] with hand unencumbered and replicated by [8]). Wrist rotation has been proposed in mobile interaction as a motion gesture to control a smartwatch [8], an interaction leveraged by several other researchers in designing systems for interaction within two-dimensional and three-dimensional spaces (e.g. [18, 32]. In the domain of smartphone input, wrist rotation has been used to define a motion gesture that can act as a gestural delimiter [35] and wrist rotation motion gestures have been commercially adopted by Motorolla smartphones as the *MotoAction* to invoke the camera app. Also, the TWuiST system [30] uses proprioceptive displacement of a smartphone, including wrist rotation, for smartphone-based interaction, and Crossnan et al. [8] studies the use of wrist rotation for command selection in sitting and walking mobile contexts. There were some work has been done on the VR menu with waist rotation [11, 14], but they all need a CD gain factor for mapping the limited waist rotation to 360°. That is less optimal for a rehearsal-based shortcut technique, which would break the connection between visual and physical memory.

2.4 Bi-manual Command Selection

For bi-manual interaction, there exists both asynchronous bi-manual interaction [15] and synchronous bi-manual interaction [23]. 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]. Leganchuk et. al. explored that bimanual input [24] has the benefits of enhanced time-motion efficiency from increased DoF and reduced cognitive load compared with traditional unimanual solutions.

3 DRUM MENU

This section outlines the motivation and design of the Drum Menu, introducing three variants: Joystick, Stroke, and Pointing-based Drum Menus.

3.1 Input Modality in VR Controllers

To make full use of the potential of VR controllers, we first analyze the main input modalities of these controllers that can be used to facilitate command access. Most commercial VR controllers comprise the following inputs:

- Boolean input: Button, on-touch/release event of buttons, joystick, touchpad, and triggers
- 1D input: Trigger
- 2D input: Joystick, touchpad
- 3D input: Position, direction

Radial menus require 2D input for angular selection and a boolean input for confirmation. For instance, pen input relies on (x, y) coordinates to calculate direction and pen-lift for confirmation. In VR, 2D input can come directly from joysticks/touchpads or be derived from 3D position/tilt. Confirmation can be triggered via buttons or by transforming 1D input (e.g., trigger pressure level).

Based on these inputs and user intuitiveness, we designed the following Drum Menu variants:

- **Joystick Drum Menu:** Joystick for direction + on-touch/release for confirmation, which is a natural way to select menu items for console or VR games.
- **Stroke Drum Menu:** Position for direction + button for confirmation, which is adopted from traditional marking menu.
- **Pointing Drum Menu:** Tilt angle for direction + button for confirmation, which is similar to ray casting in VR.

3.2 User Interactions with Different Drum Menu Variants

Each Drum Menu follows a similar flow (*i.e.*, activate, select, and confirm), but with different input modalities.

Joystick Drum Menu: 2a, the joystick is touched to activate the menu, which appears after 333 ms if still being touched. The menu is attached to the center top of the controllers, parallel to the user and vertical to the ground. Users select items by rotating the joystick, confirming by release. In the real interface (Figure 1a), joystick rotation will lead to the rotation of the white bar in corresponding controllers. The selected item will be highlighted in yellow, which applies to the rest.

Stroke Drum Menu: As shown in Figure 2b, the Stroke Drum Menu is similar to the original marking menu. A button (lower button on the controller, A or X for Oculus controllers, applies to the Pointing Drum Menu as well) is pressed to activate the menu, which appears after 333 ms. Users draw a directional stroke in the air, confirmed upon button release. The menu is displayed 10 cm in front of the controller to avoid occlusion. The stroke is mapped to a 2D plane that is vertical to the ground and parallel to the VR HMD.

Pointing Drum Menu: As shown in Figure 2c, a button is pressed to activate the menu, which appears after 333 ms. The center of the menu is positioned at the same height as the controller and 10 cm in front of it. It is positioned parallel to the HMD. The user rotates the controller in the pitch and yaw dimensions, and a ray is cast from the controller to help visualize the item selection. The selection is confirmed by releasing the button.

3.3 Novice and Expert Modes for Bimanual Drum Menus

All Drum Menus support novice and expert modes, differing in activation, selection, and display.

Bimanual novice mode: Users activate the first-level menu on the non-dominant controller. After 333 ms, the radial menu appears, and selection leads to the second-level menu on the dominant controller. The final command is confirmed upon selecting an item from this second-level menu.

Bimanual expert mode: A user activates the first-level menu on the non-dominant hand controller and quickly performs the same actions as in novice mode within a short period, without the menu appearing. The second-level menu is set up in real time based on the first-level selection, without explicitly confirming the first-level selection. The user then activates the second-level menu on the dominant hand controller and quickly selects an item without the menu appearing.

Users can also perform two actions simultaneously, which is even faster. We define simultaneous as the gap between the confirmation of two hands is less than 500ms, which also covers the case where the second-level menu is selected faster than the first-level menu shortly, as they operate independently.

4 EXPERIMENT

We condcuted an experiment to compare the performance of the three Drum Menu techniques as well as their unimanual version in two different layouts (4-item and 8-item).

4.1 Participants and Apparatus

We recruited 12 right-handed participants, ages 23 to 31, 6 women and 6 men. Participants were recruited from a local university and a local community via mailing lists. They received $20 e-gift card for their time and effort. This study was conducted both in person and remotely. The experimental software was run on Oculus Quest 2 with Unity game engine version 2019.4.40f.

4.2 Procedure

Each participant first identified their dominant hand and then watched an instructional video detailing the technique. Once they understood the technique, they began a practice block with 8 trials, either in the 8-item layout or the 4-item layout. Each trial started with two arrow prompts (Figure 4 a) indicating the correct directions: the left arrow corresponded to the first level, and the right arrow corresponded to the second level. We avoided using word prompts to minimize the learning effect and focus on expert-level performance, as our goal was to use the marking menu as a shortcut mechanism in VR. Since all techniques were radial menus, using the same prompts was a fair approach across all techniques.

Participants then selected the item corresponding to the prompt. After each selection, feedback appeared in the air: a green checkmark for a correct selection or a red "x" for an incorrect one. Incorrect selections required participants to repeat the trial until they achieved the correct response.

Upon completing the practice block, participants advanced to the main experiment, which included 2 blocks of 16 trials each. Breaks were permitted between blocks. After finishing all trials

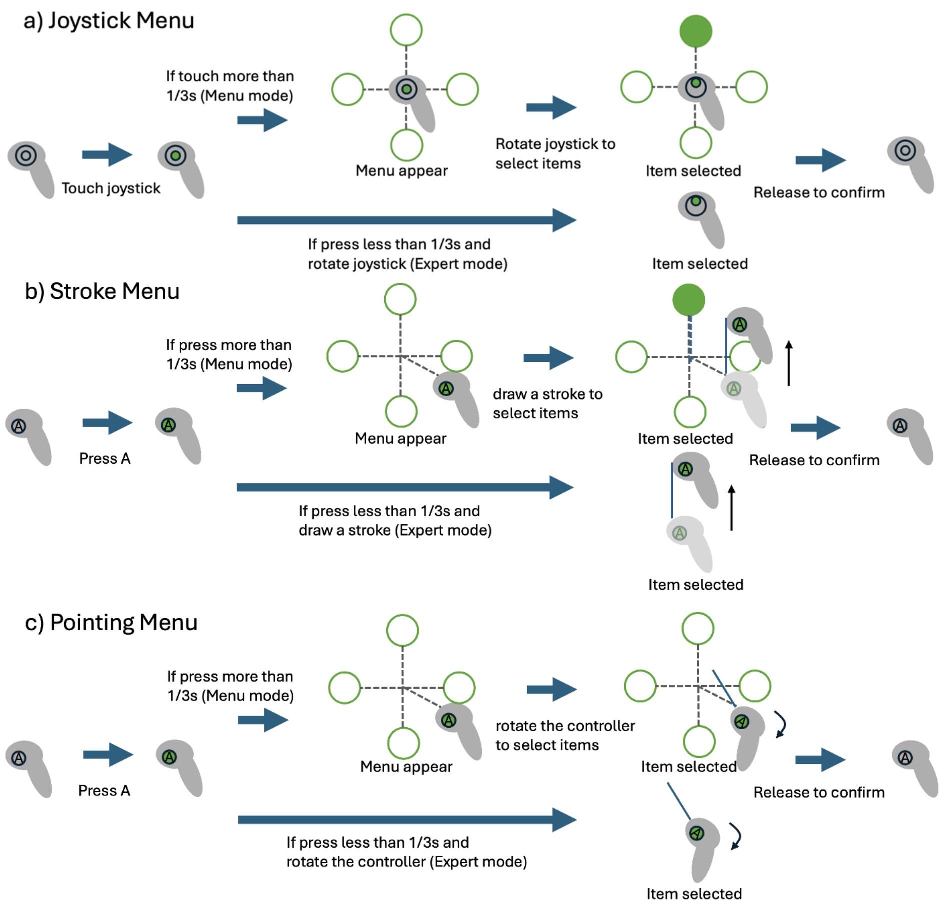


Figure 2: Workflows of the three Drum Menus.

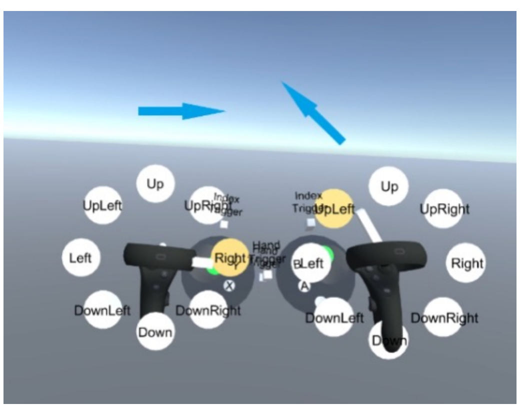


Figure 3: 8-item layout for Drum Menus.

for a specific menu condition and layout, participants completed a questionnaire before transitioning to the next layout. Once both layouts were completed, they proceeded to the next technique. After testing all techniques, the researcher conducted a semi-structured interview with the participants, inquiring about their experiences with each technique across different layouts. They were also asked about their favorite using different numbers of hands and layouts.

Each study session lasted approximately 1 hour, including breaks.

4.3 Experimental Design

We adopted a within-subjects design with two independent variables:

- TECHNIQUE (6 levels: Bimanual-Joystick, Bimanual-Stroke, Bimanual-Rotation, Unimanual-Joystick, Unimanual-Stroke, Unimanual-Rotation),
- LAYOUT (2 levels: 4-item, 8-item) (see Figures 1 and 3).

Combinations of these factors are abbreviated as 4-Bi-Joy, 8-Uni-Str, etc. Trial sequences were counterbalanced via a Latin square.

Stimuli Selection: For the 4-item layout, all 16 possible items (4 directions × 4 sub-directions) were used. For the 8-item layout, following Kurtenbach's marking menu paradigm [22], we selected 16 items evenly distributed across on-axis (Up, Down, Left, Right) and off-axis (UpLeft, UpRight, DownLeft, DownRight) directions. Four items were randomly chosen from each of four category combinations (on-on, on-off, off-on, off-off), ensuring equal representation across techniques.

Data Collection: Each participant completed 6 (Techniques) × 2 (Layouts) × 2 (Blocks) × 16 (Trials) = 384 trials.

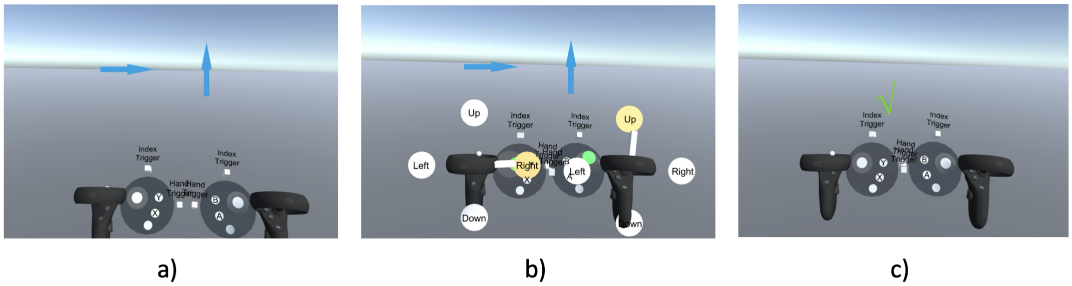


Figure 4: Task workflow: a) Two arrow appear in mid-air; b) Participant selects target item; (c) Selection confirmation.

Dependent Varaibles: We used the following measures in our study.

- *Task Completion Time*: Duration from prompt appearance to selection confirmation.
- *Error Rate*: Percentage of trials with more than one incorrect selection.
- *Total Hand Movement Distance*: Cumulative translation distance of both controllers.
- *Total Hand Rotation Angle*: Aggregate rotation angle of both hands.
- *Bimanual Asynchronize Time*: Temporal gap between first and second level confirmations in bimanual conditions. It could give us an idea if the participants are using the bimanual drum menu synchronously or asynchronously.
- *Preference*: Participants' favorite technique under different numbers of items and layouts.

Additionally, subjective workload was assessed through the NASA-TLX questionnaire, providing six subscale scores: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration.

5 RESULTS

For each combination of TECHNIQUE and LAYOUT, task completion times more than 3 standard deviations from the mean time were excluded as outliers. In total, 68 trials (1.4%) were removed.

In the analysis below a TECHNIQUE × LAYOUT Repeated Measure two-way ANOVA with combined TECHNIQUE × LAYOUT Tukey HSD post hoc tests was used, unless noted otherwise. Normality was corrected using a Box-Cox transform. When the assumption of sphericity was violated, degrees of freedom were corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$).

5.1 Task Completion Time

In general, there was a significant main effect for TECHNIQUE on task completion time ($F_{5,55} = 5.04$, $p < .001$, $\eta^2_G = 0.10$), LAYOUT ($F_{1,11} = 36.85$, $p < .0001$, $\eta^2_G = 0.39$) and TECHNIQUE × LAYOUT ($F_{5,55} = 11.99$, $p < .0001$, $\eta^2_G = 0.17$).

Figure 5 shows the task completion time for the 4-item layout. A post hoc test revealed that Bimanual variants were faster than their Unimanual variants. Bimanual Joystick (M = 947.8 ms) was the fastest one but not significantly faster than Bimanual Stroke (M = 1034.1 ms). That could be because the 4-item layout had enough space for users to select each item, and bimanual interaction allowed

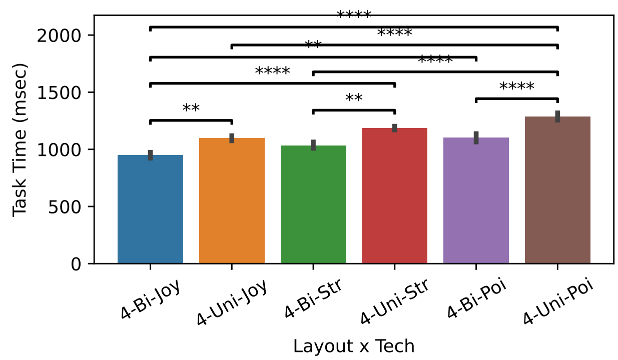


Figure 5: Task completion time for 4-item layout.

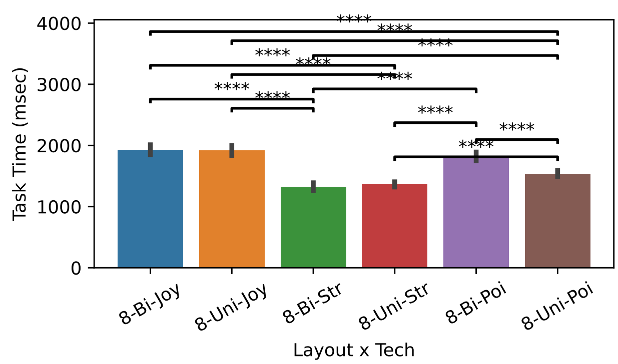


Figure 6: Task completion time for 8-item layout.

users to choose parallelly items without waiting for the first level to be selected.

Figure 6 shows the task completion time for the 8-item layout. A post hoc test revealed that only for Pointing technique Bimanual variants were slower than the Unimanual variants. Bimanual Stroke (M = 1322.2 ms) and Unimanual Stroke (M = 1359.8 ms) were the fastest ones. That could be mainly because the angle of each item is too small, which makes the users feel less confident in selecting without a visual search. However, if users draw the stroke long enough, they can still select accurately eyes-free, which turned out to be faster than the other variants.

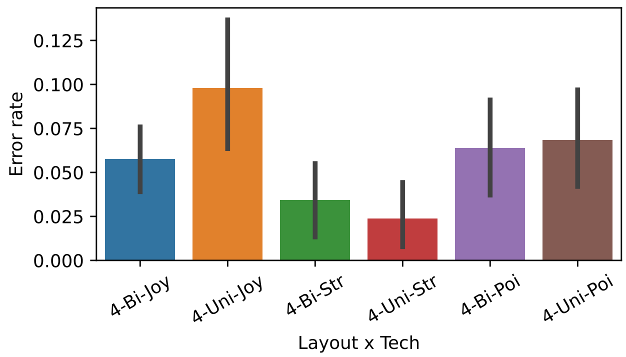


Figure 7: Error rate for 4-item layout.

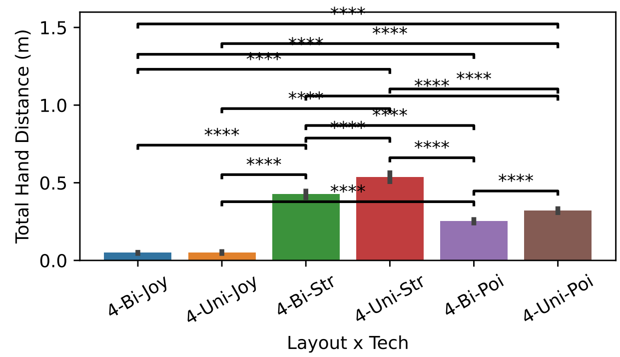


Figure 9: Total Hand Movement Distance for 4-item layout.

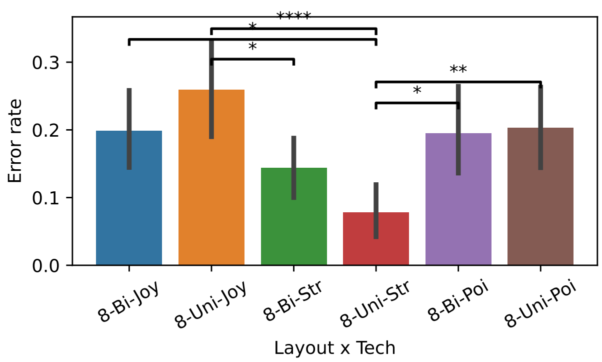


Figure 8: Error rate for 8-item layout.

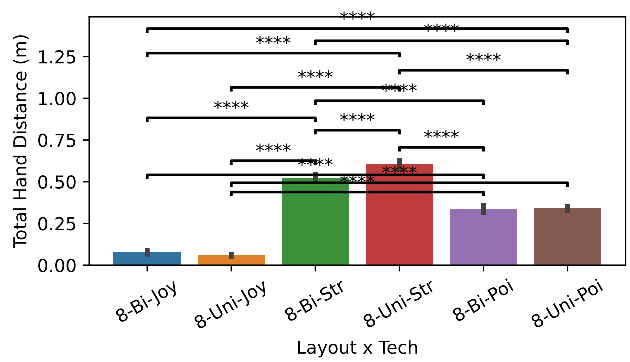


Figure 10: Total Hand Movement Distance for 8-item layout.

5.2 Error Rate

In general, there was a significant main effect for TECHNIQUE on error rate ($F_{5,55}$ = 5.05, p < .0001, η^2_G = 0.15), LAYOUT ($F_{1,11}$ = 42.62, p < .001, η^2_G = 0.30).

Figure 8 shows the error rate for the 4-item layout. All of them are less than 0.1. A post hoc test revealed that no significant error rate difference was found among all techniques. Figure 8 shows the error rate for the 8-item layout. A post hoc test revealed that Stroke variants had the lowest error rate (0.14 for Bimanual, 0.078 for unimanual). The rest are all around or higher than 0.2. The same reason as observed in task completion time: if the stroke drawn is long enough, users can still select accurately without the menu showing up.

5.3 Total Hand Movement Distance

Total hand movement distance represents the movement of both hands from the prompt time to the final selection, which would impact users' physical fatigue. In general, there was a significant main effect for TECHNIQUE on total hand movement distance ($F_{5,55}$ = 39.67, p < .0001, η^2_G = 0.69), LAYOUT ($F_{1,11}$ = 14.41, p < .0001, η^2_G = 0.03).

Figure 9 shows the total hand movement distance for the 4-item layout. A post hoc test revealed that Joystick variants (M = 0.048 m for Bimanual, M = 0.05 m for Unimanual) required the least hand movement. Pointing variants (M = 0.25 m for Bimanual, M = 0.32 m for Unimanual) required more distance. Stroke variants (M = 0.42 m for Bimanual, M = 0.83 m for Unimanual) moved the most. For the latter two variants, bimanual required less movement than unimanual variants.

Figure 10 shows the total hand movement distance for the 8-item layout, which is similar to the 4-item layout. A post hoc test revealed that Joystick variants (M = 0.076 m for Bimanual, M = 0.058 m for Unimanual) required the least hand movement. Pointing variants (M = 0.33 m for Bimanual, M = 0.34 m for Unimanual) required more distance. Stroke variants (M = 0.52 m for Bimanual, M = 0.60 m for Unimanual) moved the most.

5.4 Total Hand Rotation Angle

The Total Hand Rotation Angle refers to the degree of hand movement from the initial prompt to the final selection, which can influence users' physical fatigue.In general, there was a significant main effect for TECHNIQUE on total hand rotation angle ($F_{5,55}$ = 38.96, p < .0001, η^2_G = 0.60).

Figure 11 shows the total hand rotation angle for the 4-item layout. A post hoc test revealed that Joystick variants (M = 35.1 degree for Bimanual, M = 36.3 degree for Unimanual) required the least hand rotation. Stroke variants (M = 126.98 degree for Bimanual, M = 170.19 degree for Unimanual) required more rotation. Pointing variants (M = 166.92 degree for Bimanual, M = 199.20 degree for Unimanual) rotated the most.

Figure 12 shows the total hand rotation angle for the 8-item layout. A post hoc test revealed that Joystick variants (M = 34.15

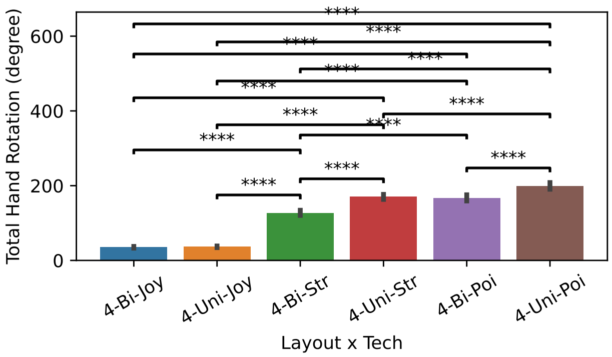


Figure 11: Total Hand Rotation Angle for 4-item layout.

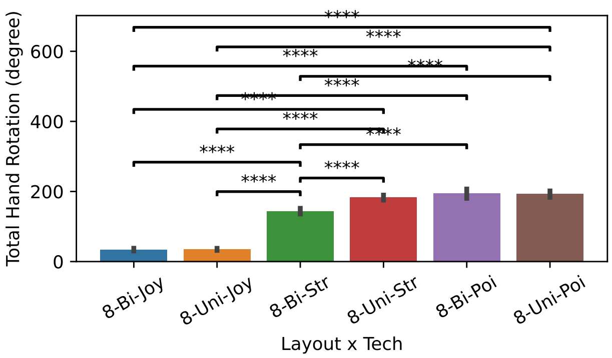


Figure 12: Total Hand Rotation Angle for 8-item layout.

degree for Bimanual, M = 34.58 degree for Unimanual) required the least hand rotation. Stroke variants (M = 143.67 degree for Bimanual, M = 182.43 degree for Unimanual) required more rotation. Pointing variants (M = 193.79 degree for Bimanual, M = 182.43 degree for Unimanual) rotated the most.

5.5 Asynchronize Time

Asynchronize Time refers to the delay between confirming the first level and confirming the second level in bimanual drum menus. The confirmation timestamp is recorded as the moment the final item is selected. This metric provides insight into whether participants are using the bimanual drum menu synchronously or asynchronously. In general, there was a significant main effect for LAYOUT on asynchronize time ($F_{1,11} = 5.68$, $p < .05$, $\eta^2_G = 0.09$).

For the 4-item layout, a post hoc test revealed that there is no difference among the three Bimanual techniques. The second level was slightly slower than the first level but almost at the same time. Figure 13 shows asynchronize time for for 8-item layout. A post hoc test revealed that Bimanual Joystick had a significantly longer asynchronize time than the others, which means the second-level confirmation time is slower.

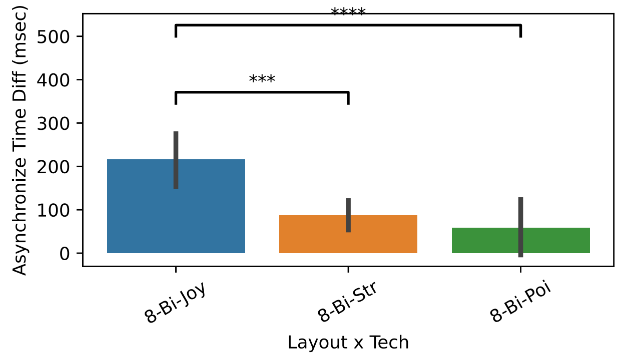


Figure 13: Asynchronize Time for 8-item layout. It represents how later is the second level confirmed than the first level.

Table 1: User preferences for different techniques under NumberOfHand and Layout.

	Joystick	Stroke	Pointing
Unimanual-4Item	11	1	0
Bimanual-4Item	10	2	0
Unimanual-8Item	2	10	0
Bimanual-8Item	3	9	0

5.6 NASA-TLX Questionnaire

We used NASA-TLX to assess the perceived workload of the techniques from participants on a 7-point Likert scale. Due to the number of techniques and questions in NASA-TLX, we highlight some key findings in the following. As shown in Figure 14, in general, the 4-item layout outperformed the 8-item layout. Moreover, 8-item Bimanual Pointing has significantly higher mental and physical demands than the others. That is mainly because the outward off-axis directions (down-left for left hand, down-right for right hand) were physically harder to reach, which requires more movement and focus to locate. Temporal pressures were similar across different conditions. For performance and effort questions, 8-item Bimanual Joystick and 8-item Bimanual Pointing had the lowest score. For frustration, the differences were not significant.

5.7 Preference Ranking

Table 1 shows that Joystick is preferred for the 4-item layout due to less physical movement, less task completion time, and less error rate. While Stroke is favored for the 8-item layout mainly because of its lower error rate compared with other techniques, even though it requires more movement than Joystick. Pointing is not preferred by any participants. It aligned with the results of Task Completion Time and Error Rate.

6 DISCUSSION

6.1 Design Implications

In the following, we extend the discussion on the differences between many key factors characterizing the Drum Menu techniques. We derived these implications by coordinating our quantitative results with the interviews with the participants as well as our observations during the study.

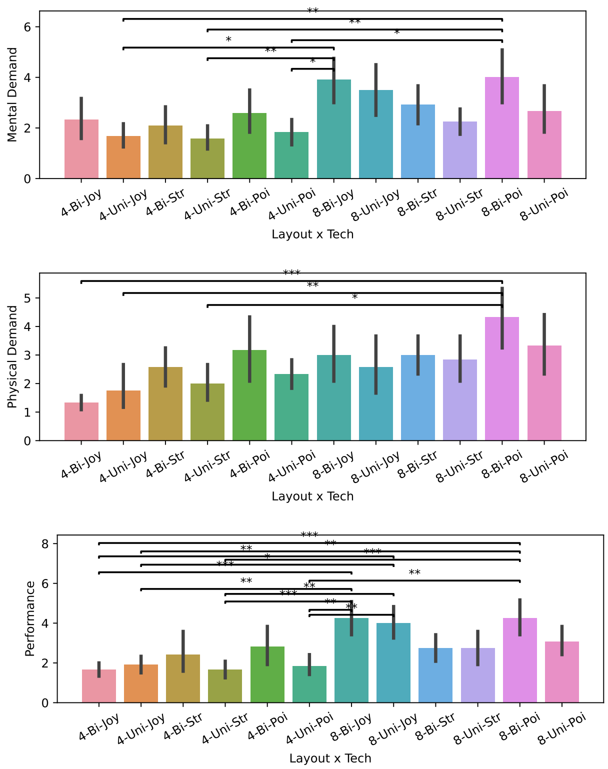


Figure 14: NASA-TLX Results

6.1.1 4-item v.s. 8-item layout. The quantitative results reveal the 8-item layout had a higher task completion time and error rate than the 4-item layouts. All participants reported that the 8-item layout was very difficult to select without carefully looking at the menu. One reason is that having more items reduces the range of each item, leading users to accidentally select neighboring items. Participants also reported that off-axis items (UpLeft, UpRight, DownLeft, DownRight) were harder to select than on-axis items. They thought they were moving to a safe angle to select the off-axis items but often ended up selecting the on-axis items instead. Due to the high error rate, participants had to activate the menu and visually locate the item, which resulted in less confidence and longer task completion times. Only the Stroke variants were slightly better in terms of error rate. It was primarily because it allowed a larger movement range than the others. When participants moved their hands further, it was easier for them to locate the correct angle without accidentally selecting nearby items. “Moving further (using the Stroke Drum Menu) makes me feel more confident to select the correct range even without looking at the menu. For the others, I have to look at the menu.” (P5) But the error rate was still high compared to the 4-item layout.

6.1.2 Bimanual v.s. Unimanual. Given that our goal is to explore the potential of the marking menu as a shortcut, we focused more on the 4-item layout. For this layout, the bimanual variants were consistently faster than the unimanual variants, with no significant difference in error rates. Most participants preferred the bimanual approach, as it allowed them to access two levels simultaneously, saving time. One participant (P10) noted that the combined effort of two hands performing individual actions was less demanding than one hand performing two actions. However, 3 participants (P6, P9, P10)mentioned that the bimanual variants required extra mental effort to coordinate both hands. “I know I can select together, but sometimes it’s hard for me to coordinate two hands together.” (P6)

6.1.3 Joystick v.s. Stroke. v.s. Pointing. In general, the Joystick and Stroke variants were the most favored. The quantitative results also revealed a similar result, as Joystick had a shorter task completion time and Stroke had a lower error rate. Participants found it challenging to rotate to downward angles or outward angles (e.g., pointing right for the right hand, pointing left for the left hand), which required greater physical effort. Remapping the angles to more comfortable positions could possibly mitigate this issue. The Joystick was preferred for the 4-item layout due to its minimal physical demand (11 out of 12 participants for unimanual, 10 out of 12 for bimanual). However, the joystick may experience jitter issues upon release, as it can bounce back or shift angles if released too quickly. This can be mitigated by ensuring selection only occurs while the joystick is being touched or requiring it to be pushed beyond a minimum threshold. However, in practice, hardware signal inaccuracies or latency may prevent this logic from completely eliminating the jitter issue. Additionally, the buttons themselves should also be taken into consideration, since there are only limited buttons in VR controllers and might also be occupied by other functionalities, like teleporting, which limits its use case. Stroke was preferred in the 8-item layout because of its lower error rate (10 out of 12 participants for unimanual, 9 out of 12 for bimanual), as discussed in the previous section regarding the 4-item v.s. 8-item layout.

6.1.4 Design Recommendation. In comparison to the existing desktop marking menus [4, 22, 39, 40], a key advantage of the Drum Menu lies in its support for bimanual input, enabling users to access commands across two levels simultaneously. Given the result, bimanual input helps users access commands faster.

As a conclusion of the discussion above, for shortcut purposes, we recommend the 4-item layout with the Bimanual Joystick Drum Menu due to its short task completion time and lower physical effort. In real-world applications, the joystick is [13] still a common way to activate pie menus. But if joysticks have been used by other functions, Stroke variants could also be a good choice.

When an 8-item layout is required to accommodate a larger command set, we recommend either the Bimanual or Unimanual Stroke Drum Menu. These variants provide the necessary flexibility to handle more commands while maintaining reasonable usability. However, it is important to note that the increased complexity of an 8-item layout may demand a higher level of visual engagement from users, as they may need to rely more on visual feedback to accurately select the desired command. This trade-off between command density and user attention should be carefully evaluated based on the specific context of use. This study primarily focuses on evaluating expert-level command selection time, which provides valuable insights into the efficiency of the proposed techniques. However, it is important to recognize that increasing the number of menu items may introduce additional challenges, such as memorization issues [39].

We recommend that designers carefully evaluate the complexity of the task and the cognitive load imposed on users when selecting an appropriate layout. For simpler tasks or applications with a limited command set, a 4-item layout may strike an optimal balance between efficiency and usability, minimizing the need for extensive memorization while maintaining fast selection times. On the other hand, for more complex tasks requiring a larger command set, an 8-item layout might be justified, but designers should consider implementing additional visual cues, progressive disclosure techniques, or training mechanisms to support user learning and retention.

Ultimately, the choice of layout should align with the specific requirements of the application, the frequency of use, and the expected user expertise. Designers should also consider conducting user studies to assess memorability and error rates across different layouts, ensuring that the chosen design not only supports expert-level performance but also accommodates the needs of all users, regardless of their familiarity with the system. This holistic approach will help create more intuitive and effective interaction designs that enhance the overall user experience.

6.2 Limitations and Future Work

In our study, we used arrows as prompts to reduce the learning effect and minimize experiment time. However, this approach only measured performance in a scenario where all menu locations were fully memorized. As a result, we could not assess how the learning effect would manifest for users who gradually familiarize themselves with the Drum Menus over time. This limitation restricts our ability to generalize the findings to real-world usage, where users may not have prior knowledge of all item locations. Future studies should investigate the learning curve associated with Drum Menus, exploring how performance evolves as users become more proficient with both unimanual and bimanual input methods in various menu layouts.

Moreover, we discovered some 6-item layout marking menus. We did not include this in our experiment because there were already too many conditions. However, a 6-item layout may still be worth exploring, given the significant performance gap between the 4-item and 8-item layouts. Participants could consistently perform expert behavior for the 4-item layout but not for the 8-item layout due to the high error rate and the resulting lack of confidence. We are curious about how the 6-item layout would perform for each Drum Menu variant.

Third, we only tested with the two-level menu. However, a three-level menu was also explored in the past [40]. Since the 4-Item layout only contains $4 \times 4 = 16$ items in total, the three-level menu could expand this to $4 \times 4 \times 4 = 64$ items. However, the interaction design may need adjustment and memorability should also be taken into consideration.

Lastly, in this study, we did not investigate the experience of transit. More studies may needed to further examine how users who are already familiar with the unimanual menu would transition the experience to bimanual variants. It could be faster because of the existing knowledge, or slower due to the inertia. We would like to start with the unimanual Drum Menu and then switch to the bimanual Drum Menu and compare the performance. Additionally, we only recruited 12 participants in the experiment. Although we reported the effect size, the result could be strengthened with more studies in the future.

7 CONCLUSION

In this paper, we present the Drum menu with six different techniques. We investigate three different ways to invoke the marking menu including Joystick, Stoke, and Pointing. Unimanual and bimanual variants were also discovered. We tested their performance against two layouts, the 4-item layout and the 8-item layout. In general, we recommend the Bimanual Joystick variant with the 4-item layout as the shortcut mechanism in VR due to its short task completion time, low error rate, and low physical movement.

ACKNOWLEDGMENTS

This work was supported by NSERC. We sincerely thank our study participants for their time and valuable feedback in evaluating our techniques. We also appreciate the insightful suggestions provided by the reviewers, which helped improve this work.

REFERENCES

[1] 2014. Tilt Brush. https://www.tiltbrush.com/
[2] Sunggeun Ahn, Stephanie Santosa, Mark Parent, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2021. StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.
[3] 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* (Denver, Colorado, USA) *(CHI EA '17)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/3027063.3052959
[4] 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* (Napoli, Italy) *(AVI '08)*. Association for Computing Machinery, New York, NY, USA, 15–22. https://doi.org/10.1145/1385569.1385575
[5] 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* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 627–636. https://doi.org/10.1145/1240624.1240723
[6] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting novice to expert transitions in user interfaces. *ACM Computing Surveys (CSUR)* 47, 2 (2014), 1–36.
[7] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (nov 2014), 36 pages. https://doi.org/10.1145/2659796
[8] 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* (Amsterdam, The Netherlands) *(MobileHCI '08)*. Association for Computing Machinery, New York, NY, USA, 435–438. https://doi.org/10.1145/1409240.1409307
[9] 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. https://doi.org/10.1080/14640748308402133
[10] Raimund Dachselt and Anett Hübner. 2007. Three-dimensional menus: A survey and taxonomy. *Computers & Graphics* 31, 1 (2007), 53–65.
[11] Gabriel Di Domenico, Natan Luiz Paetzhold Berwaldt, Mauren W. D'Avila, and Cesar Tadeu Pozzer. 2023. Radial Menu for Virtual Reality Based on Wrist Rotation. *Proceedings of the 25th Symposium on Virtual and Augmented Reality* (2023). https://api.semanticscholar.org/CorpusID:266788047
[12] Blender Foundation. [n. d.]. Home of the blender project - free and open 3D creation software. https://www.blender.org/
[13] Epic Games. n.d.. XR - Unreal Engine. https://www.unrealengine.com/en-US/xr. Accessed: 2024-10-17.
[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] 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.

[17] 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* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 8, 9 pages. https://doi.org/10.1145/3338286.3340119

[18] 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.

[19] Kenrick Kin, Bjoern Hartmann, and Maneesh Agrawala. 2011. Two-handed marking menus for multitouch devices. *ACM Trans. Comput. Hum. Interact.* 18 (2011), 16:1–16:23. https://api.semanticscholar.org/CorpusID:5926356

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

[21] Rick Komerska and Colin Ware. 2004. A study of haptic linear and pie menus in a 3D fish tank VR environment. *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.* (2004), 224–231. https://api.semanticscholar.org/CorpusID:15234153

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

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

[24] Andrea Leganchuk, Shumin Zhai, and William A.S. Buxton. 1998. Manual and cognitive benefits of two-handed input: an experimental study. *ACM Trans. Comput. Hum. Interact.* 5 (1998), 326–359. https://api.semanticscholar.org/CorpusID:6871433

[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] Wenmin Li, Xueyi Wan, Yanwei Shi, Nailang Yao, Ci Wang, and Zaifeng Gao. 2021. Depth and breadth of pie menus for mid-air gesture interaction. *International Journal of Human–Computer Interaction* 37, 2 (2021), 131–140.

[27] Zhen Li, Joannes Chan, Joshua Walton, Hrvoje Benko, Daniel Wigdor, and Michael Glueck. 2021. Armstrong: An empirical examination of pointing at non-dominant arm-anchored UIs in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[28] 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.

[29] 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.

[30] 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.

[31] Martin Mundt and Tintu Mathew. 2020. An evaluation of pie menus for system control in virtual reality. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*. 1–8.

[32] 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.

[33] 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* (Brisbane, QLD, Australia) *(VRCAI '19)*. Association for Computing Machinery, New York, NY, USA, Article 30, 9 pages. https://doi.org/10.1145/3359997.3365693

[34] Gang Ren and Eamonn O'Neill. 2012. 3D marking menu selection with freehand gestures. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 61–68.

[35] 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.

[36] Johann Wentzel, Matthew Lakier, Jeremy Hartmann, Falah Shazib, Géry Casiez, and Daniel Vogel. 2024. A Comparison of Virtual Reality Menu Archetypes: Raycasting, Direct Input, and Marking Menus. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[37] 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.

[38] 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* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 42, 13 pages. https://doi.org/10.1145/3173574.3173616

[39] 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* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1077–1086. https://doi.org/10.1145/1124772.1124933

[40] 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* (Santa Fe, NM, USA) *(UIST '04)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1029632.1029639

[41] 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* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 448, 12 pages. https://doi.org/10.1145/3290605.3300678