Experimental Analysis of Single Mode Switching Techniques in Augmented Reality

Jesse Smith*
University of Florida

Isaac Wang†
University of Florida

Julia Woodward‡
University of Florida

Jaime Ruiz§
University of Florida

Abstract

This paper presents an empirical evaluation of mode switching techniques for Augmented Reality (AR) headsets. We conducted a quantitative analysis exploring five techniques for switching between modes: a hardware button press, 3D virtual button press, non-preferred hand, reach depth, and voice. Results from our study support the benefits of non-preferred mode switching, showing non-preferred and depth mode switching to be faster than voice and the virtual button techniques. Depth, however, had significantly more errors compared to the other techniques. Our work lays a foundation for developers to design new mode switching techniques and guides the design of current hardware solutions around choosing techniques that best compliment application use.

Keywords: augmented reality; mode switching; mode errors

Index Terms: H.5.2. User Interfaces – Interaction Techniques

1 Introduction

Augmented Reality (AR) headsets (such as the Microsoft HoloLens [35]) enable users to interact with digital objects projected onto the physical world through gaze, speech, and hand gestures. Many interaction techniques in AR (such as hand gestures) can be used to enable multiple software states (e.g., both selection and manipulation [52]). For example, consider touching a virtual object; this could trigger its selection, or manipulate its properties through translation or rotation. Since one interaction technique can be mapped to multiple software states, there is a need to evaluate techniques that perform operations manipulating the state (i.e., mode) of the software alongside actions manipulating the digital content. The changing of software state (referred to as *mode switching*) has shown to be time-consuming and error prone in stylus [28,31,48,49] and touch-based interfaces [55]. As such, researchers have explored new mode switching techniques that aim to fluidly allow both command and input [16,21,50] and have conducted formal studies to analyze their human performance.

Li et al. [31] explored five different techniques for mode switching with stylus-based interfaces and concluded that the use of the non-preferred hand to switch modes is the most efficient and highest-rated. Surale et al. [55] extended this work to touch-based interfaces by performing an empirical study comparing six touch-based mode switching techniques. They also found that the use of the non-preferred hand to switch modes was highly rated, although using two-fingers was the most efficient.

While mode switching techniques have extensively been formally evaluated for pen and touch-based interfaces, we are unaware of any work that empirically evaluates mode switching for use in AR headsets. These headsets differ from device-based AR

*jd.smith@ufl.edu, †wangi@ufl.edu, ‡julia.woodward@ufl.edu, §jaime.ruiz@ufl.edu

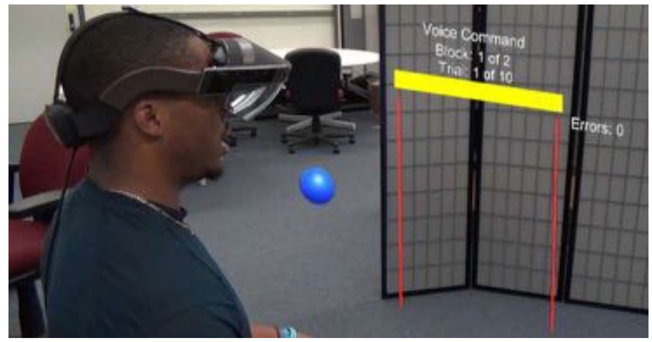

Figure 1: The design of the virtual environment and experimental setup for our study.

(e.g., Apple's ARKit [2] or Google's ARCore [14]) by allowing users to interact directly (i.e., hands-free) with virtual objects without needing a handheld display (e.g., smartphone). The goal of this work is to provide an initial exploration into examining different hands-free mode switching techniques in an AR interface.

In this paper, we evaluated five mode switching techniques for use in AR headsets: pressing a physical button on the AR headset, pressing a 3D virtual button, gesturing with the non-preferred hand, varying reaching distance (i.e., depth), and giving a voice command. These techniques are derived from methods currently utilized in commercial devices and those used in pen and touch-based mode switching studies. Our formal evaluation provides the following results and insights:

(1) The non-preferred hand technique is good for mode switching in terms of preference and efficiency.
(2) Using depth and a virtual button are perceived as being inaccurate and difficult to use.
(3) The voice and hardware button techniques are very accurate but have limitations (e.g., slower than non-preferred hand).

Our work provides an initial evaluation of mode switching techniques for AR headsets. This evaluation could guide design considerations of new and existing mode switching techniques in AR and inform decisions around choosing which techniques best compliment application and hardware capabilities.

2 Related Work

Many researchers have explored interaction techniques that aim to allow both command and input in an efficient and error-free manner [21,50,55,59]. We focus our review of prior work on three categories: 1) improving mode switching efficiency, 2) evaluating current mode switching techniques, and 3) exploring mode switching in AR interfaces.

2.1 Increasing Mode Switching Efficiency and Access

Prior work has examined interaction techniques that do not require mode switching, as well as making existing mode switching techniques more efficient in pen-based systems. Saund and Lank [50] proposed a technique that minimizes the need to switch modes by inferring a user's intent from the properties of the pen trajectory and the context of the interaction to automatically switch the system state. If the intent is ambiguous, the user is offered a choice to change the mode. Hinckley et al. [21] demonstrated the use of a post-gesture delimiter for combining the acts of selection, command activation, and direct manipulation.

Several researchers have examined the use of the non-preferred hand as an efficient interaction technique to switch modes [23,28,31,48,49]. Non-preferred hand mode switching is an asymmetric bimanual task – a two-handed task in which each hand has a different role. The non-preferred (i.e., non-dominant) hand controls the state, or mode, of the interface, while the preferred hand performs the moded action. Li et al. [31] studied five different existing mode switching techniques and concluded that, of the five techniques, non-preferred hand performed the best based upon the metrics of speed (fastest), error rate (second lowest), and user preference (most preferred). Hinckley et al. [23] introduced Springboard which demonstrated how to increase the availability of multiple modes when a user holds a button with their non-preferred hand. Lank et al. [28] found that by allowing a mode switch to occur concurrently with performing the task gesture, there is no significant difference in the time taken to perform a gesture with and a gesture without a mode switch; thus, resulting in a "cost-free mode switch". Ruiz and Lank [49] extended Lank et al.'s work by examining the scalability of adding modes with the non-preferred hand technique, which eventually led to the development of a temporal model (i.e., model describing the time costs associated with the process of switching modes) [48]. Some other methods proposed to efficiently switch modes in pen-based systems have included using the space above a tablet device [16], using the pressure [45] or tilt [6] of the stylus, using a touch sensor below the palm of the writing hand [51], and using gestures and different grips on the barrel of the stylus [24,53].

More recently, researchers have extended work on pen-based user interfaces to take advantage of new touch-enabled displays. Several researchers have examined how the non-preferred hand can be used on a touch surface to: activate different modes for a pen in the dominant hand [9,58], explore methods for using the non-preferred hand holding a tablet to switch modes [11,59], and enable different interactions based on pressure from the dominant or non-preferred hand [20,46]. An example of this interaction is GripSense [13], which uses both pressure and device grip to support differentiating input. Besides pressure, other properties of finger input such as the number of fingers [55,58], contact size [8], slight rolling movements [47], and the specific part of the finger touching the display [19,55] have been examined to support mode switching.

Prior work in 3D environments has focused on examining selection tasks [15,41,52] and manipulation tasks [40,43,44,52]. Within these tasks, there have been different interaction methods, including gaze and click [41], ray casting [4,38,62], and natural gesturing [12,42]. AR user interfaces have other interaction methods that could be adapted as mode switching techniques, such as voice or an external input device.

Voice interactions have been studied previously in conjunction with gesture input [7,25]. In particular, Bolt [7] designed a study to examine the functionality of using speech and gesture to select and manipulate objects on a 2D screen. This study utilized a speech recognizer that listened for certain commands (i.e., pronouns) to activate a manipulation. Bolt found that the conjunction of speech and gesture allowed users to reference objects on a screen more precisely. Some examples of external input devices include gaming controllers, virtual reality (VR) controllers, and mobile devices [5,18,33,60]. Gallo et al. [33] experimented with the visualization and interaction of 3D reconstructed organs in a virtual environment. The study aimed to find more natural methods with existing interaction techniques to allow users to interface with virtual human anatomy. The authors utilized the button layout and motion sensing capabilities of the Nintendo Wiimote [39] as an intuitive input device for manipulation and pointing tasks.

To our knowledge, researchers have not directly explored mode switching in AR. We are interested in designing a study to generalize some of these techniques to evaluate their efficiency as mode switching techniques in AR interfaces.

2.2 Evaluating and Modeling Mode Switching

Evaluation has been an important part in exploring the effectiveness of new proposed mode switching techniques [55]. Dillon et al.'s [10] "subtraction technique" is a common methodology for comparing mode switching techniques [31,55]. In the subtraction technique a trial consists of multiple crossing tasks that alternate between requiring and not requiring a mode switch. The temporal cost of a mode switch is then determined by subtracting the time required to perform the task without a mode switch from the moded task (i.e., the task requiring a mode switch). Another technique that is used to examine the temporal cost of mode switching is *line cutting*. In line cutting, a trial consists of one task, bissecting two lines using either a moded or unmoded gesture (Fig. 1). The temporal cost of the mode switch is determined by comparing the mean time to complete the moded tasks to the mean time of the unmoded tasks. Line cutting tasks have been extensively used to examine non-preferred hand mode switching on pen-tablet interfaces [28,48,49], and was better suited for our AR mode switching study to minimize arm fatigue associated with prolonged, repetitive mid-air gestures.

2.3 Mode Switching in AR

We are unaware of any prior work that has systematically examined and compared mode switching techniques for AR headsets. Surale et al. [55] found differing technique performances when moving from stylus to touch interfaces. The researchers found that in stylus interfaces the non-preferred hand technique still performed well in terms of speed and preference, but using the two-finger interaction was fastest overall. This result showed a variation in technique performance between the two interfaces. In addition, AR differs from pen and touch interfaces in that users do not physically hold a stylus or touch a screen to manipulate content. Instead, they rely primarily on mid-air gestures that lack haptic feedback. It is likely that technique performance differences will exist when evaluating them in an AR environment due to these interaction challenges and previous performance variations.

3 Mode Switching Techniques

We examined five mode switching techniques; three from interaction techniques commonly used in AR (hardware button, virtual button, and voice) and two techniques adapted from pen and touch-based mode switching studies (non-preferred hand and depth). These techniques are illustrated in Figure 2.

3.1 Hardware Button Press

A physical button press is a common input method as shown in prior work [28,31,48,49]. AR and VR interfaces (e.g., Microsoft HoloLens) typically use keypads or handheld controllers to navigate menus and manipulate display content. However, in our study we specifically focused on hands-free interaction, so we did not consider handheld devices for our mode switching techniques. Instead, we utilized the physical buttons on an AR headset as a mode switching technique (Fig. 2a). The buttons on the headset normally display information and control the volume, however, we repurposed them to alternate between the mode states. The volume-

Figure 2: The five mode switching techniques used: (a) hardware button, (b) virtual button, (c) non-preferred hand, (d) depth, and (e) voice.

up button switched to the moded state, the volume-down button switched to the unmoded state, and the display information button toggled between the two states. Once a participant pressed and released a button, the action was performed. The participants were allowed to freely interact with the three buttons; the buttons were located on the right-hand side of the headset.

3.2 Virtual 3D Button Press

Virtual buttons are commonly used in user interfaces to initiate an action, change display/content settings, and manipulate objects. We implemented a 3D button to toggle between the different mode states (Fig. 2b). Leap Motion [29], a company specializing in hand and finger interactions in 3D environments, recommends that one finger targets in virtual environments should be >20mm [30], therefore a full-handed target should be >100mm. Following this recommendation, we designed the button to have a top-face surface dimension of 200 x 100mm. We placed the virtual button in a stationary position 250mm below the virtual sphere. To interact with the button, the participant would reach out and press the button as if it was a physical button. Since there is no tactile feedback when pressing the virtual button, we implemented two visual cues: the button would visually depress with the participant's interaction, and the button would change color when fully pressed.

3.3 Non-Preferred Hand Mode Switching

Non-preferred hand interactions have received extensive exposure in previous studies [11,22,26,28,34,59]. We include this interaction as it allows the separation of tasks between the two hands; manipulating content with one hand while using the non-preferred hand to change the manipulation state. Previous studies in pen and touch-based interfaces utilized the non-preferred hand to press a button on the interface. Since AR headsets are equipped with depth sensors, we used these sensors to recognize a hand pose from the participant's non-preferred hand as the mode switching technique (Fig. 2c). Participants would interact with the virtual object with one hand, while using the other hand (i.e., non-preferred hand) to change the mode state by posing an "open-hand" for switching to the moded state or "closed-fist" for switching to the unmoded state.

3.4 Depth-Based Mode Switching

We examined reach depth, which was adapted from pressure-based mode switching techniques used in pen and touch-based interfaces (Fig. 2d) [31,55]. Adapting pressure for depth-based interactions in AR allows us to evaluate its performance and compare it to previously studied interfaces. Li et al. [31] showed that pressure could be a promising mode switching technique, but also found that it is not as effective as other techniques (e.g., non-preferred hand).

For our study, we implemented a uniform depth threshold, which is consistent with Li et al.'s uniform pressure threshold for pen. Participants would have to reach 100 mm into a virtual sphere of diameter 200 mm to trigger a mode switch. Once the object is in the ideal mode it can be grabbed and translated. We chose this reach distance because 1) the participant's hand stays in contact with the object, maintaining the perception of grabbing the sphere, and 2) this distance was deep enough to differentiate between the two mode states while minimizing unintended grab events triggered by extended contact with the virtual object.

3.5 Voice-Triggered Mode Switching

Voice is a common method for smart assistants, for example Amazon Alexa [1] and Apple Siri [3]. User interfaces like the Microsoft HoloLens [35] and Windows mixed reality headsets [36] utilize these assistants to manipulate content such as opening menus, adjusting input/output controls, and making selections. In our study, we utilized this method by allowing the participant to say their desired mode (Fig. 2e). Microsoft Windows integrated dictation recognizer [37] was used for its easy implementation and quick keyword recognition.

4 Experiment

Our experimental procedure analyzing our five mode switching techniques was influenced by Lank et al.'s [28] line cutting design.

4.1 Participants

The participants in our study included 20 adults, ages 18 to 25 ($M = 21.04$, $SD = 2.22$). Seven of the participants were female, and six of the participants were left-handed. We removed data from one participant due to equipment failure. All of the participants were students recruited from the University of Florida, four of which had previous experience with AR devices. Participants either received extra credit for a course they were enrolled in or voluntarily participated without compensation. Our protocol was approved by our Institutional Review Board.

4.2 Apparatus

The experiment was conducted using the Meta 2 AR headset. The headset was tethered to a PC and features an LCD display with a resolution of 2560 × 1440 px at a 72 Hz refresh rate, projected onto a partially mirrored, transparent lens which allows for a 90-degree field of view.

A Blue Yeti microphone was used to capture high-quality sound from the user during the voice-triggered mode switching technique. The integrated buttons on the Meta 2 headset were used as input for the hardware button mode switching technique, and the other mode switching techniques (e.g., depth) utilized the Meta 2's depth camera to recognize user hand pose input. The virtual environment was developed using Unity [57], a game development platform.

4.3 Task

We based our experimental task design on Lank et al.'s [28] line cutting task. In a line cutting task with pen-based interfaces participants had to draw a line to bisect the two lines. For our study, we replaced drawing with object manipulation to adhere more to current applications in AR and VR. Our virtual environment consisted of a virtual object (i.e., sphere) located to the left of two vertical panels (Fig. 1). The two vertical panels were placed 800 mm apart. A horizontal color panel, located above the two vertical panels, indicated the intended mode of the object (blue or yellow).

During the experiment, participants were asked to grab and move the virtual sphere from left to right, bisecting the two vertical bars, and then release the sphere in the area to the right of the rightmost bar. Prior to moving the sphere, the participants had to either leave the sphere unmoded (blue) or change the sphere to its moded state (yellow) based on the color of the horizontal color panel. This color changing as a mode switching task has been utilized in previous mode switching studies to represent the simplification of changing between common modes in the studied interface [31,55].

4.4 Design and Procedure

Our experiment is a repeated measures design (i.e., the participants used all five mode switching techniques). The mode switching techniques were counterbalanced across participants using a Latin Square. Participants were seated to ensure that body fatigue, from standing and gesturing in midair, did not influence our results.

For each mode switching technique, participants were given a brief overview of how to use the technique to activate a mode switch. Participants were instructed to perform all techniques with one hand, except for the non-preferred hand technique, which adheres to prior work [32,55]. Once participants understood how to use the technique, they were then given a practice block to get familiar with the technique in the virtual environment. The practice block consisted of 10 untimed trials (5 unmoded and 5 moded) and was not included in analysis. When the participants finished the practice block, they would then complete two experimental blocks, each with 10 trials (5 unmoded and 5 moded). The task was organized into two blocks in order to give participants breaks and to preserve the Latin Square design. We recorded errors and the time taken to complete each trial.

The procedure for each trial is as follows: 1) a three-second countdown, 2) once the sphere and horizontal color panel appeared the participant would change the mode of the sphere (i.e., blue or yellow) based on the color panel, 3) the participant would grab the object and move it across the two vertical panels, 4) the participant would release the sphere to the right of the rightmost vertical panel, and 5) the sphere and color panel would disappear which would initiate the countdown for the next trial. Between each block (i.e., 10 trials) the participants were given at least a five-second break, and between each mode switching technique (i.e., 2 blocks) the participants took a break to complete a subjective survey.

We had a total of 1900 trials (19 participants × 5 mode switching techniques × 2 blocks × 2 modes × 5 trials per mode). Out of the 1900 trials, 950 required a mode switch.

5 Results

We analyzed the five mode switching techniques by examining learning effects, error rates, trial times, and subjective preferences.

5.1 Learning Effects

To ensure that our participants did not show significant improvements in performance as the trials progressed, we examined the mode switching times and combined error rates for each technique per block. To do this we ran a repeated measures ANOVA which found no significant main effects of technique × block for both metrics. This confirmed that our participants kept their performance consistent across blocks.

5.2 Error Analysis

Similar to prior work [31,55], we recorded three types of errors: mode errors, crossing errors, and out-of-target errors. *Mode errors* occurred when the wrong mode was selected at the initiation of the grab event (unmoded when moded needed and vice versa). A *crossing error* occurs when the grabbed object does not bisect both vertical panels before being released. This measure captures errors related to crossing accuracy and participants intentionally aborting the line cutting task. An *out-of-target error* is recorded when the

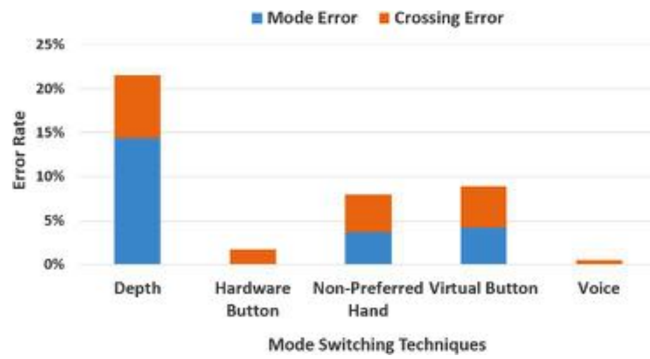

Figure 3: Error rate per mode switching technique. Error rate contains both mode errors and crossing errors.

grabbed object is released before bisecting the leftmost vertical panel, which is related to equipment related issues.

We removed all trials containing out-of-target errors so that our analysis would not be influenced by equipment issues. This included 127 trials (6.7% of all trials), 72 of which were moded trials (7.6% of moded trials). We also removed unmoded trials from our analysis to isolate errors associated with trials requiring a mode switch (Fig. 3). A Shapiro-Wilks test on the error rate distributions showed that the data was severely skewed, so we applied the Aligned Rank Transform [61]. We used the transformed data for analysis, but the mean error rates presented in this paper are the actual measured values.

5.2.1 Mode Error Rate

We analyzed the average mode error rate per technique. A repeated measures ANOVA found a significant main effect of technique on the mode error rates ($F_{4,72} = 12.39, p < .001$). Mode error rates from lowest to highest are: voice and hardware button (0%), non-preferred hand (3.68%), virtual button (4.21%), and depth (14.37%). We ran a post-hoc comparison and found that participants using depth made significantly more errors.

5.2.2 Crossing Error Rate

We analyzed the average crossing error rate per mode switching technique. A repeated measures ANOVA found significant main effects of technique on crossing error rates ($F_{4,72} = 2.85, p < .05$). Crossing error rates from lowest to highest are: voice (0.53%), hardware button (1.70%), non-preferred hand (4.27%), virtual button (4.74%), and depth (7.15%). Post-hoc comparisons showed that participants made significantly fewer crossing errors using voice. Further analyzing crossing errors, we found that 52% of these errors were made after a mode error. This means that about half the time participants failed to complete the trial when realizing the mode was not changed correctly.

5.3 Trial Time Analysis

A complete trial is composed of two actions: 1) switching the mode of a virtual object (i.e. *mode switch*), and 2) translating the virtual object across two vertical lines (i.e. *crossing*). *Mode switching time* measured the time between the presentation of the desired mode (displayed by the color panel) to the object grab event. *Crossing time* measured the duration between the object grab event to the object release event within a trial. There are 10 trials (i.e., crossing) per block (100 per participant), half of which require a mode switch (50 mode switches per participant).

We examined the time to complete each task. A Shapiro-Wilks test on the average mode switching and crossing times showed that both distributions were non-normal. We applied a log-transform to

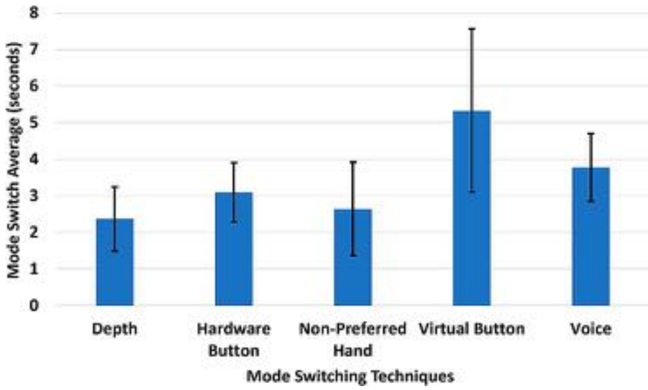

Figure 4: Average mode switch time, in seconds, per technique. Error bars represent standard deviation.

the distributions and used the transformed data for analysis, but the mean task times presented in this paper are actual measured values.

5.3.1 Mode Switching Time

We analyzed the average mode switching time per technique (Fig. 4). A repeated measures ANOVA found a significant main effect of technique on the mode switching times ($F_{4,72} = 18.98, p < .001$). The average mode switching times, in seconds, from fastest to slowest are: depth (2.35s), non-preferred hand (2.63s), hardware button (3.09s), voice (3.77s), virtual button (5.24s). Post-hoc comparisons showed that virtual button and voice were significantly slower than the rest of the mode switching techniques. The measured values show that virtual button is about 2x slower and voice is about 1.4x slower than the rest of the techniques.

5.3.2 Crossing Time

We analyzed mean crossing times to ensure that the crossing task did not affect our mode switching analysis; no significant variation in average crossing times between techniques should exist. A repeated measures ANOVA did not find any significant main effect of technique on the crossing times ($\mathrm{F}_{4,72} = 1.93, \mathrm{p} > .05$).

5.4 Subjective Preference Analysis

We conducted a subjective survey, derived from Li et al.'s [31] pen mode switching study. The survey provided insight on: ease of learning, ease of use, accuracy, speed, eye fatigue and arm fatigue. All of the mode switching techniques were ranked on a 5-point Likert scale by category, with 1 being the worst (e.g., hardest to learn) and 5 being the best (e.g., easiest to learn).

We performed a Shapiro-Wilks test on the subjective rating distributions which showed that the data was severely skewed. An Aligned Rank Transform correction was applied to the distributions. We used the transformed data for analysis, but the mean preference ratings presented in this paper are actual measured values. For overall subjective ratings we found a significant main effect of technique ($\mathrm{F}_{4,72} = 4.21, \mathrm{p} < .01$). We expected low preference for the virtual button due to high errors rates and slow mode switching times. Post hoc comparisons confirmed our expectation showing that both virtual button and depth were preferred significantly less than voice, hardware button, and non-preferred hand (rated highest to lowest respectively).

We further analyzed virtual button and depth to determine what factors led to these techniques being least preferred. A repeated measures ANOVA on each subjective category found significant main effects of technique for learning ($\mathrm{F}_{4,72} = 5.52, \mathrm{p} < .001$), ease of use ($\mathrm{F}_{4,72} = 6.79, \mathrm{p} < .001$), and accuracy ($\mathrm{F}_{4,72} = 9.08, \mathrm{p} < .001$). No significant main effects were found for perceived speed, eye fatigue, and arm fatigue. Post-hoc comparisons showed that depth was significantly harder to learn, and both depth and virtual button were significantly less accurate and harder to use.

We further examine the five different mode switching techniques and how our analysis situates in prior work in the discussion below.

6 Discussion

Our results show that hardware button, voice, and non-preferred hand are good mode switching techniques for AR interfaces.

6.1 Techniques for Mode Switching in AR

We discuss each technique from the most promising to the least promising. For each technique, we look at their strengths and weaknesses with respect to mode switching performance and subjective rating. We situate our findings in prior work and discuss how further improvements in AR can impact mode switching.

6.1.1 Non-preferred Hand

We found that the non-preferred hand as a mode switching technique performed well overall. Prior work exploring the use of non-preferred hand for mode switching in other domains (e.g., pen and touch) has found it to be an efficient technique [23,23,28,31,48,49,55], reducing the cost of mode switching by allowing both hands to work in conjunction [28,48,49]. Non-preferred hand also had the third lowest mode error rate, at only 3.68%. We believe the largest contributor to errors for this technique was the hand recognition of the Meta 2 headset, which would sometimes fail to detect the presence of the second hand or misrecognize whether the hand was open or not. However, as the error rate was small, recognition was only an occasional issue and future updates to the headset could improve the accuracy.

We note the advantage of this technique to support multiple modes. In its current implementation, recognizing hand poses by counting fingers allows the application to support up to 6 modes (one mode for each number of fingers held up, from 0-5 fingers). Beyond finger counting, AR can utilize more complex hand recognition to capture more hand poses (e.g., thumbs-up) and mid-air gestures (e.g. swipe right) performed by the non-preferred hand, allowing for a larger set of mode mappings. Due to its mode support and good performance overall, we believe non-preferred hand is a viable technique for mode switching in AR.

6.1.2 Hardware Button

The use of the hardware buttons on the headset to switch between modes was very effective, with no errors and the third fastest mode switching times. The hardware button technique was rated second-highest in terms of preference. Its main advantages come from the physical nature of the buttons: they are fully tactile, making them easy to interact with, and they are oriented around the head, which has been shown to improve discoverability without needing to see the buttons [17]. It was also difficult to accidentally press the buttons, and they triggered reliably, resulting in low error rates.

Table 1: Mean subjective preference for each technique.

	Depth	Hardware Button	Non-Preferred Hand	Virtual Button	Voice
Learning	4.11	4.79	4.79	4.47	4.79
Use	3.58	4.47	4.32	3.26	4.47
Accuracy	3.16	4.47	4.16	2.95	4.21
Speed	4.11	4.21	4.47	3.74	4.05
Eye Fatigue	4.32	4.53	4.53	4.37	4.42
Arm Fatigue	4.26	3.95	3.84	4.47	4.53
Sum	**23.54**	**26.42**	**26.11**	**23.26**	**26.47**

The placement of the buttons on the AR headset may have limited how fast participants could switch modes. As the buttons were placed on the right side of the headset, participants would need to move their hand up to the headset, press a button, and then move their hand all the way to the sphere to continue the task. Another drawback is that this technique can only support as many modes as there are buttons. Supporting more modes would require more external buttons, or more clicks per action on integrated buttons. Overall, hardware button was a reliable technique.

6.1.3 Voice

Voice was a very effective mode switching technique, with no mode errors and the highest preference rating. The major drawback of this technique is speed, performing second-worst in terms of mode switching time, significantly slower than depth, non-preferred hand, and hardware button. This was primarily due to the speed and accuracy of automatic speech recognition, as the system would need to wait for the user to stop speaking and recognize that the mode keyword was spoken before switching the mode. Participants would need to wait for the mode to switch, so it is likely that they would have waited to verify that the mode was switched, possibly explaining the low error rate. It would also have been difficult to accidentally trigger a mode switch, which may have contributed to the low error rate.

The challenges with using voice interactions become prevalent in noisy environments (e.g., factory), or in environments where speaking is not appropriate (e.g., library). However, in ideal conditions, voice offers a number of advantages. For one, it can support a virtually unlimited number of modes, as long as there are unique words/phrases to trigger each mode. Voice also does not require the user to do anything with their hands, leaving them free to continue to interact with virtual content even when switching modes. As long as speech recognition continues to improve, voice can become a suitable form of mode switching in AR.

6.1.4 Depth

Using depth showed promise in terms of how quickly it can initiate a mode switch. Our results showed that depth was the fastest (2.35 seconds to change the mode). This was likely because participants did not need to perform any grossly different actions to switch modes; all that was required was reaching further into the sphere.

Depth had the largest number of mode errors overall, with a 14.37% error rate, and was the least preferred technique. The large error rate was likely due to difficulties in determining how far into the sphere a user needed to reach. This was similar to prior work on pressure-based mode switching for pen and touch interaction [45,55], in which users found it difficult to gauge how much pressure to apply to a stylus or touchscreen to switch modes. Visual cues may help alleviate this issue. Depth was also perceived to be significantly harder to learn, harder to use, and less accurate in terms of participants' subjective ratings. Participants mentioned that since this technique was not considered a "typical" user input modality, it was harder to learn and "difficult to use properly."

We observed that participants' hand orientation combined with the contact with the sphere caused the hand recognition to falsely register a grab. This would have caused unintended mode errors if a grab was incorrectly detected before the user was able to properly switch the mode. Participants suggested that being able to initiate the mode switch after grabbing the sphere would improve this technique, alleviating some of the issues described above.

6.1.5 Virtual Button

The virtual button was the worst performing technique out of the five. Our results showed that virtual button had the slowest average mode switching times, the second-highest mode error rate, and the lowest subjective preference ratings.

Based on comments from participants and our observations of their interactions, we believe that the poor performance was due to a number of factors. Participants mentioned how it was difficult to judge how far away the button was in space and that it was difficult to know how far the button needed to be pushed down in order to trigger. This would increase the mode switching time, as participants would need to first reach for the button, ensure that they are correctly interacting with it, and then fully depress the button to toggle a mode switch. Additionally, the physics of the button sometimes caused the mode switch to fail; participants reported that they often had difficulty initiating the button which may have contributed to mode switching time and mode error rate.

The difficulties of interacting with the button appear to be related to the capabilities of the AR headset, both in rendering virtual objects and recognizing a user's hands relative to those objects. Deficiencies in both areas would make it difficult to consistently interact with virtual objects, or judge how far an object lies. Providing users with accurate depth perception has been shown to be a challenge for AR systems [27,56], and hand recognition using depth sensors also remains an ongoing challenge [54]. A lack of haptic feedback when pushing the button may also have made it difficult for users to determine if they were correctly interacting with the button. Improving the rendering and recognition of AR headsets would be required before a virtual button can be an effective and reliable option for mode switching.

7 Limitations and Future Work

We view our work as an exploratory study into mode switching in AR. In the previous section we discussed advantages and disadvantages of each mode switching technique in this study. While our study is not exhaustive and focused on hands-free interaction methods, we see the opportunity to investigate other interaction techniques for mode switching (e.g., in-air gestures). An extension of this work should also consider handheld devices (e.g., controller) which are typically included with headsets such as the Microsoft HoloLens. With handheld controllers, users could potentially improve mode switching times, when compared to pressing a button on the headset.

Though our work provides insight on the effectiveness of evaluated mode switching techniques in AR, we understand that our study evaluated the performance of a single mode switch. Our analysis is an initial examination of mode switching in AR which thoroughly evaluates the efficiency of an isolated mode switch for each technique. However, AR interfaces have multiple working states and provide users with different ways of interaction. Further evaluation to validate our findings for multiple mode switching, and exploring more unique interaction techniques in AR, would scale our work to a more realistic use of the interface. We then would be able to model this scalability with the temporal cost of switching between modes analyzed by Ruiz and Lank [49]. Similar to other studies on mode switching [31,55], our evaluation was completed using a controlled task. Future work should investigate mode switching in more realistic and uncontrolled tasks.

8 Conclusions

We examined five mode switching techniques for use in an AR interface: hardware button, virtual button, non-preferred hand, depth, and voice. We found that non-preferred hand performed the best as a mode switching technique in terms of preference, efficiency, and mode support. A virtual button should not be implemented as a mode switching technique in AR because it had the longest mode switching time and was not preferred. Both hardware button and voice performed well (e.g., low error rate); however, the hardware button is limited in the number of modes it can support and voice is limited by software recognizers and the noise level of the environment. Depth had the fastest mode switching time, but also had the highest error rate. We recommend improving depth by developing a depth indicator to assist the participants in knowing how far they have reached. Our results help to inform hardware and software design for future AR interfaces.

ACKNOWLEDGMENTS

We thank the McKnight Doctoral Fellowship program for making this work possible and our reviewers for their valuable feedback.

REFERENCES

1. Amazon. Amazon Alexa. Retrieved September 21, 2018 from https://alexa.amazon.com/spa/index.html
2. Apple. ARKit. Retrieved September 21, 2018 from https://developer.apple.com/arkit/
3. Apple. Siri. Retrieved September 21, 2018 from https://www.apple.com/siri/
4. Ferran Argelaguet and Carlos Andujar. 2009. Visual feedback techniques for virtual pointing on stereoscopic displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology - VRST '09*, 163. https://doi.org/10.1145/1643928.1643966
5. Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. 2016. NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, 717–728. https://doi.org/10.1145/2984511.2984526
6. Xiaojun Bi, Tomer Moscovich, Gonzalo Ramos, Ravin Balakrishnan, and Ken Hinckley. 2008. An exploration of pen rolling for pen-based interaction. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*, 191. https://doi.org/10.1145/1449715.1449745
7. Richard A. Bolt. 1980. "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques - SIGGRAPH '80*, 262–270. https://doi.org/10.1145/800250.807503
8. Sebastian Boring, David Ledo, Xiang "Anthony" Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The fat thumb: using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*, 39. https://doi.org/10.1145/2371574.2371582
9. Peter Brandl, Clifton Forlines, Daniel Wigdor, Michael Haller, and Chia Shen. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. In *Proceedings of the working conference on Advanced visual interfaces - AVI '08*, 154. https://doi.org/10.1145/1385569.1385595
10. R. F. Dillon, Jeff D. Edey, and Jo W. Tombaugh. 1990. Measuring the true cost of command selection: techniques and results. In *Proceedings of the SIGCHI conference on Human factors in computing systems Empowering people - CHI '90*, 19–26. https://doi.org/10.1145/97243.97247
11. Cédric Foucault, Manfred Micaux, David Bonnet, and Michel Beaudouin-Lafon. 2014. SPad: a bimanual interaction technique for productivity applications on multi-touch tablets. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA '14*, 1879–1884. https://doi.org/10.1145/2559206.2581277
12. Markus Funk, Mareike Kritzler, and Florian Michahelles. 2017. HoloLens is more than air Tap: natural and intuitive interaction with holograms. In *Proceedings of the Seventh International Conference on the Internet of Things - IoT '17*, 1–2. https://doi.org/10.1145/3131542.3140267
13. Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, 545. https://doi.org/10.1145/2380116.2380184
14. Google. ARCore. Retrieved September 21, 2018 from https://developers.google.com/ar/discover/
15. Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*, 3. https://doi.org/10.1145/1166253.1166257
16. Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, 861. https://doi.org/10.1145/1124772.1124898
17. Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, 49–60. https://doi.org/10.1145/2984511.2984576
18. Gerwin de Haan, Eric J. Griffith, and Frits H. Post. 2008. Using the Wii Balance Board as a low-cost VR interaction device. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology - VRST '08*, 289. https://doi.org/10.1145/1450579.1450657
19. Chris Harrison, Julia Schwarz, and Scott E. Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 627. https://doi.org/10.1145/2047196.2047279
20. Seongkook Heo and Geehyuk Lee. 2012. ForceDrag: using pressure as a touch input modifier. In *Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI '12*, 204–207. https://doi.org/10.1145/2414536.2414572
21. Ken Hinckley, Patrick Baudisch, Gonzalo Ramos, and Francois Guimbretiere. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, 451–460. http://doi.acm.org/10.1145/1054972.1055035
22. Ken Hinckley, Mary Czerwinski, and Mike Sinclair. 1998. Interaction and modeling techniques for desktop two-handed input. In *Proceedings of the 11th annual ACM symposium on User interface software and technology - UIST '98*, 49–58. https://doi.org/10.1145/288392.288572
23. Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. 2006. The springboard: multiple modes in one spring-loaded control. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, 181. https://doi.org/10.1145/1124772.1124801
24. Ken Hinckley, Andrew Wilson, Michel Pahud, Hrvoje Benko, Pourang Irani, François Guimbretière, Marcel Gavriliu, Xiang "Anthony" Chen, Fabrice Matulic, and William Buxton. 2014. Sensing techniques for tablet+stylus interaction. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*, 605–614. https://doi.org/10.1145/2642918.2647379
25. Sylvia Irawati, Scott Green, Mark Billinghurst, Andreas Duenser, and Heedong Ko. 2006. "Move the couch where?" : developing an augmented reality multimodal interface. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, 183–186. https://doi.org/10.1109/ISMAR.2006.297812
26. Paul Kabbash, William Buxton, and Abigail Sellen. 1994. Two-handed input in a compound task. In *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94*, 417–423. https://doi.org/10.1145/191666.191808
27. E. Kruijff, J. E. Swan, and S. Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, 3–12. https://doi.org/10.1109/ISMAR.2010.5643530
28. Edward Lank, Jaime Ruiz, and William Cowan. 2006. Concurrent bimanual stylus interaction: a study of non-preferred hand mode manipulation. In *Proceedings of Graphics Interface 2006* (GI '06), 17–24. Retrieved from http://dl.acm.org/citation.cfm?id=1143079.1143083
29. Leap Motion, Inc. Leap Motion. Retrieved September 21, 2018 from https://www.leapmotion.com/

30. LeapMotion. 2015. VR Design Best Practices. *LeapMotion*. Retrieved December 14, 2018 from https://medium.com/@LeapMotion/vr-design-best-practices-bb889c2dc70
31. Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, 461. https://doi.org/10.1145/1054972.1055036
32. Yang Li, Ken Hinckley, Zhiwei Guan, and James A. Landay. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, 461. https://doi.org/10.1145/1054972.1055036
33. Luigi Gallo, Giuseppe De Pietro, and Ivana Marra. 2008. 3D interaction with volumetric medical data: experiencing the Wiimote. In *Proceedings of the 1st international conference on Ambient media and systems*. https://doi.org/10.1145/1363163.1363177
34. Nobuyuki Matsushita, Yuji Ayatsuka, and Jun Rekimoto. 2000. Dual touch: a two-handed interface for pen-based PDAs. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00*, 211–212. https://doi.org/10.1145/354401.354774
35. Microsoft. Microsoft HoloLens. Retrieved September 21, 2018 from https://www.microsoft.com/en-us/hololens
36. Microsoft. Windows Mixed Reality. Retrieved September 21, 2018 from https://www.microsoft.com/en-us/windows/windows-mixed-reality
37. Microsoft. Continuous Dictation. Retrieved September 21, 2018 from https://docs.microsoft.com/en-us/windows/uwp/design/input/enable-continuous-dictation
38. Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. 1997. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH '97*, 19–26. https://doi.org/10.1145/258734.258747
39. Nintendo. Wii Accessories. Retrieved September 21, 2018 from https://www.nintendo.com/wiiu/accessories
40. Noritaka Osawa. 2008. Two-Handed and One-Handed Techniques for Precise and Efficient Manipulation in Immersive Virtual Environments. In *Advances in Visual Computing*, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Paolo Remagnino, Fatih Porikli, Jörg Peters, James Klosowski, Laura Arns, Yu Ka Chun, Theresa-Marie Rhyne and Laura Monroe (eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 987–997. https://doi.org/10.1007/978-3-540-89639-5_94
41. Hyung Min Park, Seok Han Lee, and Jong Soo Choi. 2008. Wearable augmented reality system using gaze interaction. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, 175–176. https://doi.org/10.1109/ISMAR.2008.4637353
42. Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA '13*, 955. https://doi.org/10.1145/2468356.2468527
43. I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. 1998. Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques. *Computer Graphics Forum* 17, 3: 41–52. https://doi.org/10.1111/1467-8659.00252
44. Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology - UIST '96*, 79–80. https://doi.org/10.1145/237091.237102
45. Gonzalo Ramos, Matthew Boulos, and Ravin Balakrishnan. 2004. Pressure widgets. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, 487–494. https://doi.org/10.1145/985692.985754
46. Christian Rendl, Patrick Greindl, Kathrin Probst, Martin Behrens, and Michael Haller. 2014. Presstures: exploring pressure-sensitive multi-touch gestures on trackpads. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 431–434. https://doi.org/10.1145/2556288.2557146
47. Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 927. https://doi.org/10.1145/1518701.1518843
48. Jaime Ruiz, Andrea Bunt, and Edward Lank. 2008. A model of non-preferred hand mode switching. In *GI '08: Proceedings of graphics interface 2008*, 49–56.
49. Jaime Ruiz and Edward Lank. 2007. A study on the scalability of non-preferred hand mode manipulation. In *ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces*, 170–177. http://doi.acm.org/10.1145/1322192.1322223
50. Eric Saund and Edward Lank. 2003. Stylus input and editing without prior selection of mode. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03), 213–216. https://doi.org/10.1145/964696.964720
51. Itiro Siio and Hitomi Tsujita. 2006. Mobile interaction using paperweight metaphor. In *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06*, 1325. https://doi.org/10.1145/1125451.1125697
52. Chang Geun Song, No Jun Kwak, and Dong Hyun Jeong. 2000. Developing an efficient technique of selection and manipulation in immersive V.E. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '00*, 142. https://doi.org/10.1145/502390.502417
53. Hyunyoung Song, Hrvoje Benko, Francois Guimbretiere, Shahram Izadi, Xiang Cao, and Ken Hinckley. 2011. Grips and gestures on a multi-touch pen. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, 1323. https://doi.org/10.1145/1978942.1979138
54. J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. 2015. Depth-Based Hand Pose Estimation: Data, Methods, and Challenges. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 1868–1876. https://doi.org/10.1109/ICCV.2015.217
55. Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2017. Experimental Analysis of Mode Switching Techniques in Touch-based User Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 3267–3280. https://doi.org/10.1145/3025453.3025865
56. J. E. Swan, A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman. 2007. Egocentric depth judgments in optical, see-through augmented reality. *IEEE Transactions on Visualization and Computer Graphics* 13, 3: 429–442. https://doi.org/10.1109/TVCG.2007.1035
57. Unity Technologies. Unity. Retrieved September 21, 2018 from https://unity3d.com
58. Daniel Vogel and Géry Casiez. 2011. Conté: multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 357. https://doi.org/10.1145/2047196.2047242
59. Julie Wagner, Stéphane Huot, and Wendy Mackay. 2012. BiTouch and BiPad: designing bimanual interaction for hand-held tablets. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 2317. https://doi.org/10.1145/2207676.2208391
60. Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. 2018. Haptic Revolver: Touch, Shear, Texture, and Shape Rendering on a Reconfigurable Virtual Reality Controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, 1–12. https://doi.org/10.1145/3173574.3173660
61. Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, 143. https://doi.org/10.1145/1978942.1978963
62. Chongbin Xu, Mingquan Zhou, Dongdong Zhang, Wuyang Shui, and Zhongke Wu. 2013. Guidance rays: 3D object selection based on multi-ray in dense scenario. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry - VRCAI '13*, 91–100. https://doi.org/10.1145/2534329.2534347