EyeSeeThrough: Unifying Tool Selection and Application in Virtual Environments

Diako Mardanbegi*
Lancaster University, Lancaster, UK

Ken Pfeuffer†
Bundeswehr University Munich, Munich, Germany

Alexander Perzl‡
LMU Munich, Munich, Germany

Benedikt Mayer§
LMU Munich, Munich, Germany

Shahram Jalaliniya¶
Centennial College, Toronto, Canada

Hans Gellersen‖
Lancaster University,
Lancaster, UK

ABSTRACT

In 2D interfaces, actions are often represented by fixed tools arranged in menus, palettes, or dedicated parts of a screen, whereas 3D interfaces afford their arrangement at different depths relative to the user and the user can move them relative to each other. In this paper we introduce EyeSeeThrough as a novel interaction technique that utilises eye-tracking in VR. The user can apply an action to an intended object by visually aligning the object with the tool at the line-of-sight, and then issue a confirmation command. The underlying idea is to merge the two-step process of 1) selection of a mode in a menu and 2) applying it to a target, into one unified interaction. We present a user study where we compare the method to the baseline two-step selection. The results of our user study showed that our technique outperforms the two step selection in terms of speed and comfort. We further developed a prototype of a virtual living room to demonstrate the practicality of the proposed technique.

Index Terms: Human-centered computing—Human-centered-computing—Gestural input;

1 INTRODUCTION

3D virtual environment afford direct operation of objects using their body and hand movements. Interfaces can be designed in relation to the body to exploit the user's sense of proprioception [32]. Hand-held palettes [9, 29, 45] and cockpit menus [14] are examples of such user interfaces in which the menu items are fixed to the user's body, enabling users to remain aware of, and interact with, both the menu interface and the background objects in the scene at any time.

The application of a tool's effect to an intended target is based on a two-step process. Target objects and menus are typically presented at distinct locations in the scene. As a result, the interaction is two-fold: First, the user points at the tool in the menu and confirms the selection. Second, the user points at the object, and confirm the application of the tool. This can also be done by first pointing at and selecting the object, and secondly, pointing at the tool and confirming its application to the object. At times such as when the user's task involves frequent switching between different tools and targets, it can become a tedious and inconvenient operation. For this reason, this research explores how the two-step process can be unified into one interaction.

We introduce *EyeSeeThrough*, an interaction technique for VR that leverages eye-tracking to streamline the two-step operation by utilizing the ability to align the object and tool relative to each other
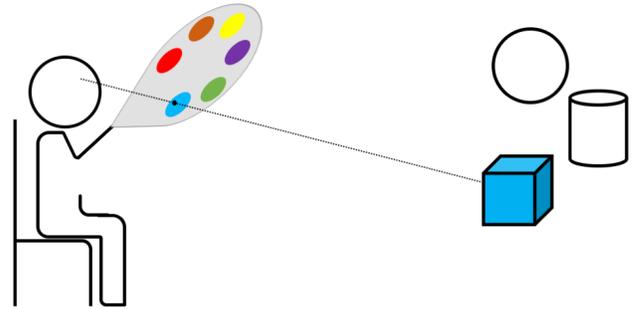
*e-mail: d.mardanbegi@lancaster.ac.uk
†e-mail: ken.pfeuffer@unibw.de
‡e-mail: alexanderperzl@gmx.de
§e-mail: benedikt_mayer@outlook.de
¶e-mail: sjalali3@my.centennialcollege.ca
‖e-mail: hwg@comp.lancs.ac.uk

Figure 1: The EyeSeeThrough technique unifies tool selection and application: the user visually aligns the menu's tool with the target in line of sight. Here the user applies the blue color to the cube object.

in 3D. In particular, users apply a tool by visually aligning it with a target, followed by a confirmation (Figure 1). The alignment is based on explicit movement of the tool displayed on a semi-transparent layer in the view relative to the target in the background, inspired by Bier et al.'s Toolglass technique for 2D desktop UIs [5]. The user simply looks at the target while positioning the tool over it. When the user confirms the alignment (e.g., by click), the 3D gaze ray determines both the tool and the target, and the action is executed in the virtual scene. Therefore, two successive selections are unified into one interaction, analogous to a shooting task where the user aligns the gun's crosshair with the target and triggers the action.

The eyes play an important role in visual alignment task. The technique infers the alignment of the tool and the object by examining the intersection of the user's gaze ray with the object and the tool in 3D. Therefore, gaze is used implicitly for the interaction. Although gaze is employed similar to a pointing mechanism, the technique doesn't require users to move their eyes between the object and the tool, but allows them to retain their visual attention on the desired object while aligning the tool with it. In other words, gaze is used for pointing towards two targets at once which we believe offloads the extra pointing step to other modalities such as hand to bring the tool towards line-of-sight.

In this paper, our goal is to better understand the underlying interaction process of EyeSeeThrough from a design and a user perspective. We first describe two variations of the technique: one where the tool palette is attached to the user's hands, and one where the menu is attached to the user's head like a cockpit interface. We then present a user study that compares the baseline two-step to our approach, to understand the effect of unifying the steps. Our study shows promising results on EyeSeeThrough. In the tested color-matching task, the user's performance improves over the baseline two-step approach, and users also preferred the technique.

Based on the findings, we explore the design space of the technique through application examples to demonstrate how the technique can be used in realistic environments. For instance, users

can leverage the cockpit interface to control smart objects such as a lamp or a TV through EyeSeeThrough. Lastly, by considering the study and applications, we explore its design space by discussing design considerations of the technique pointing towards the rich possibilities revealed by interacting with EyeSeeThrough.

2 RELATED WORK

Our work builds on three main relevant strands of prior work: (1) Gaze-based interaction in VR, (2) Interaction with menus in VR, and (3) Visual alignment for interaction.

2.1 Gaze Interaction in VR

Our study explores an interaction technique based on visual alignment that uses gaze as the main input in combination with body movements (e.g. head or hand).

Pointing and selecting objects using virtual pointer (raypointing) or flash light has been extensively explored in the context of virtual environments [20, 22, 28, 31, 37]. Raypointing through hand/arms could be subject to inaccuracy through hand jitters, eye-hand visibility mismatch conflicts [1], or the Heisenberg effect [9]. Head pointing has also been widely adopted in VR/AR user interfaces as a standard and a hands-free way of pointing at virtual objects by aligning a cross-hair in the center of the field of view with the object (e.g., Oculus Rift [1] and Microsoft HoloLens [2]).

Several studies have explored using eye gaze as an alternative means for pointing in virtual environments:

Gaze raycast: Tanriverdi and Jacob [40] proposed one of the early work on using gaze interaction in VR applications. They outlined two main reasons of why the use of gaze as an interaction means seems compelling in VR environment: 1) the users' pre-existing abilities to perform interaction with a VR interface in the absence of conventional WIMP-based commands (such as typing keywords or clicking by mouse), 2) eye gaze allows for targeting distant objects faster and with less physical effort than pointing with hand [34, 38, 42]. They confirmed the latter by conducting an experiment comparing the performance of manual and gaze input for interacting with close and distant objects in VR. Cournia et al. [11] conducted a similar study to compare the performance of eye gaze and hand pointing (ray-casting), but observing lower performance for eye gaze for distant objects. In our case, whether gaze is faster or slower for pointing, we consider eye gaze an interesting cue for identifying objects of interest in a tool application context. Beyond pointing, combinations of gaze and head/hand motion have also been studied for precise pointing [21, 27], primarily addressing the limitations of using gaze for precise pointing due to calibration errors and low accuracy.

Raycasting from the eye: Argelaguet et al. [2] showed that compared to laser rayscasting, the selection performance in cluttered environments can be significantly improved when the wrist rotations are used for changing the orientation of a virtual ray which is cast from the eye position.

Wand-gaze intersection: Zeleznik et al. 2005, proposed a suite of gaze-based interaction techniques for VR including an example where the pointing task is done by intersecting a hand-held pointing ray with an invisible plane that passes through the user's eyes and extends along the viewing direction [47].

In addition to the gaze pointing techniques, others studied gaze and eye movements for selection in VR. Piumsomboon et al. [36] introduced three different eye-gaze only techniques: a Duo-Reticles technique as an alternative to dwell-time selection, Radial Pursuit based on smooth pursuit and head-gesture-based selection using vestibulo-ocular eye movements which was previously explored for interaction with desktop user interfaces [30].

2.2 Menu Interaction in VR

Menus are widely studied for interaction in 3D environments for tasks such as navigation, selection, manipulation and system control [7, 10]. Besides differences in size, geometry and layout, menus in 3D could vary based on the position and whether they are context menus displayed around the object, menus attached to the user's arm/hand [8, 17, 31]) or head-centered menus [6, 15]. Various techniques have been proposed for interaction with menus including hand- or head-based raypointing techniques or more intuitive control based on real world gestural manipulation without controllers [23, 31, 37]. Pfeuffer et al. [34] explored a hybrid Gaze + Pinch technique that integrates eye gaze to select, and in-direct freehand gestures to manipulate targets and menus of any distance.

Commonly, menu interactions include two steps: selecting the object (the target) and a menu item, where the users have to move their visual attention from the object towards the menu items or visa versa. Instead, we investigate maintaining the visual focus on the target while positioning the tool over it.

2.3 Visual Alignment for Interaction

We identified two lines of previous work that work based on visual alignment:

Occlusion selection (Image-plane interaction): These techniques are based on the concept of aligning the user's hand or a hand-held object with the target object in the scene [16, 35]. Compared to the raycasting techniques that are based on aligning the object of interest with a pointing ray, in the occlusion selection techniques, the users align the object of interest with a hand-held aperture [16] or with their fingers [35] which are actually located at different depth than the object. For example, in the Head Crusher technique proposed by Pierce et.al. [35], the user positions her/his thumb and forefinger around the target object to select that object. These works do not leverage eye-tracking, e.g., the Head Crusher technique determines which object is between the user's fingers by casting a ray into the scene from the user camera through the point between the user's forefinger and thumb. These techniques did not consider multiple tools (actions) nor situations where multiple objects are aligned with the aperture.

Toolglass: The principle of aligning menu items with objects has been introduced before in the context of 2D graphical user interfaces by Bier et al. (1993) [5]. They used the term *See-through interface* or Toolglass which referred to a set of click-through tools arranged on a semi-transparent layer that can be moved over the application layer like a floating palette. The Toolglass was designed for use with a pair of indirect pointing devices (e.g., two mice or a mouse and a trackball), where 1) one is used to move the tool-sheet and position the tool on top of the object with the non-dominant hand, and 2) the other for positioning a cursor on top of the object and clicking through the tool with the dominant hand. The possibility of using direct-touch for moving the palette and pointing through the tool layer has also been studied on tablets [33]. Other studies have extended the metaphor of Magic Lenses (a class of see-through interfaces) to three-dimensional environments [43]. In our technique, we passively incorporate users' eye gaze to provide context to the alignment task by inferring their intention as the line of sight intersects with a tool and an object.

3 EYESEETHROUGH

EyeSeeThrough is conceived as the general technique of alignment with gaze. We first describe the EyeSeeThrough interaction technique as the general concept, and then as two concrete examples of it showing how the technique could be applied in an interaction context:
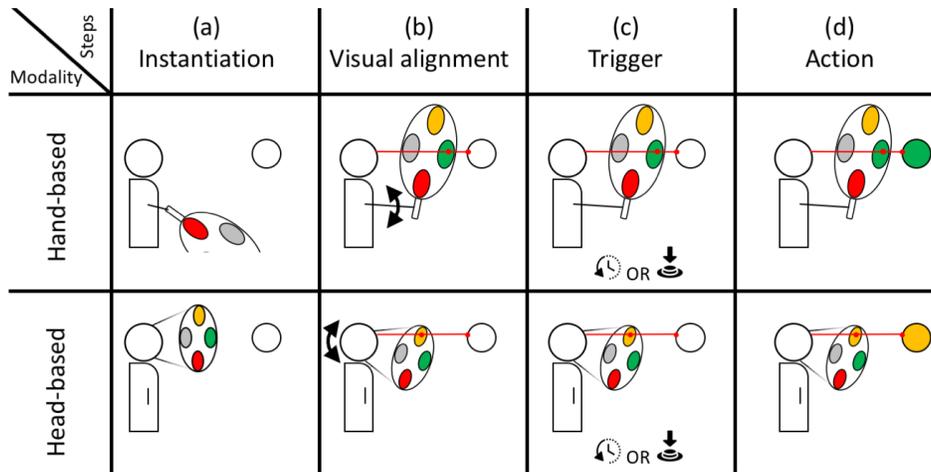
Figure 2: Illustration of 4 main steps involved in hand-based (top row) and head-based (bottom row) EyeSeeThrough.

3.1 General Technique

The operation of the EyeSeeThrough technique can be summarized into the following 4 steps inspired by the Toolglass technique presented in [4]:

1. *Instantiation step*: Instantiation step refers to when both the object of interest and the tool, are present in the user's field of view and the user can start aligning them. In many cases, the menu is already available and this step could be taken for granted.

2. *Visual alignment*: This step refers to the task of aligning the tool and the object with the line-of-sight which is typically done by looking at the object and bringing the tool towards it until it's lined up with the gaze ray.

   Gaze plays a primary role in our technique as in practice there may be multiple objects and multiple tools available to align in the user's view. The system implicitly determines the intended object/action through the gaze information. In the visual alignment step, three main components of the interaction (gaze, object, and tool) have to be aligned together. This can be done by maintaining the gaze on the interactive object while applying motion to bring the tool towards it. In general, the motion could be applied to the object, the tool, or in some cases both as they can be moved relative to each other. The user could be simply instructed to see the object through a tool.

   Visual alignment is one of the well-developed skills that people learn in their interaction with the physical world. The emerging virtual/mixed reality technology offers lots of opportunities to use body movements for directly move, manipulate and align objects in 3D [13]. In principle, different graphical objects (e.g. hand-held menus or even background objects) can be attached to different parts of the user's body and the user could move them separately exploiting the user's sense of preconception [32]. The user may also move her/his head or body in 3D space to adjust the view and to reduce the distance between the objects in the field of view.

3. *Trigger*: Confirmation that executes an action (to the object) which will be interpreted according to the tool that was looked through.

4. *Action*: A command that is executed by the system in response to a triggering event. Although, the alignment step in EyeSeeThrough could be taken as confirmation, enough for

issuing a command, in a more conservative approach (to avoid 'Midas Touch' problem [19] or perhaps false positives), the system can wait for an explicit request from the user to issue the command.

3.2 Hand-Based EyeSeeThrough

The hand-based EyeSeeThrough variant uses manual input for holding the menu and activating the mode of the menu to the target. In particular, it involves 4 steps (as illustrated in Figure 2): The system brings the menu that is attached to the handheld controller, into the view (a). The user aligns the mode of the menu with the target in the environment (b). Then, by clicking a button on the handheld controller or by maintaining the alignment for a few miliseconds (dwell time) (c), the user applies the mode to the target (d).

By using manual input to control menu and mode activation, the user has more control over their interactions. It allows the user to move the menu in and out of their view at any time. This variant is suitable for VR systems that come with handheld controllers or hand-tracked capability.

3.3 Head-Based EyeSeeThrough

The head-based EyeSeeThrough variant resembles a see-through cockpit menu where the menu is attached to the user's head. Compared to the hand-based variant, it uses head movement input for activating the mode of the menu to the target (Figure 2): The system displays the menu in the user's view (a). The user moves their head (menu) and aligns the mode of the menu with the target in the environment (b). Then, by clicking a button or by maintaining the alignment for a few miliseconds (dwell time) (c), the user applies the mode to the target (d).

This variant is more suitable for see-through displays (e.g., AR or MR) that don't come with hand-tracking capability to attach a virtually menu to the user's hand. It also allows the user to perform the interaction hands-free as the selection in our technique can be confirmed by dwell selection or a head gesture.

4 USER STUDY

The core idea of EyeSeeThrough is that it merges the two steps tool selection and application into one unified interaction. In this study, we want to investigate this aspect by comparing it to the baseline two-step approach that requires first selecting the tool and then the target (or visa verse) where it naturally requires the user to shift their visual attention away from the target of interest.

The two-step baseline is based on gaze pointing where users first point to the tool, then to the object. Gaze pointing has been
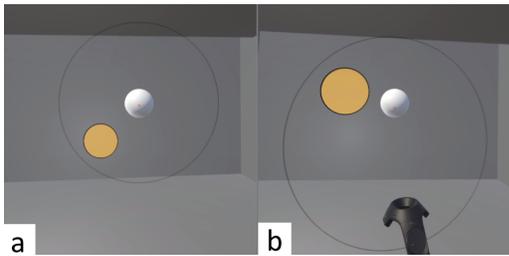
Figure 3: Screenshots from the study showing head-based (a) and hand-based (b) menus in the user's view. The screenshot does not match the actual Field-of-View of the user, however, the gray circle in the left figure indicates 80% of the user's FoV.

extensively investigated for its potential to interact in 3D environments [11, 34, 36, 40, 47].

We used an abstract coloring task of objects using a semi-transparent color tool that is inspired by Kabbash et al's Toolglass evaluation [24]. Such a task specifically allows to investigate the idea of alignment between a tool menu and targets in space.

Further, we compare the use of two different modalities for performing the visual alignment task, the *Hand-based* and *Head-based EyeSeeThrough*. The layout of the color menu was kept identical for conditions. Overall, there are 2x2 conditions: one- × two-steps, head × hand.

4.1 Participants

We recruited 12 participants (9 male and 3 female) among students from our university. 3 of them had no experience with virtual reality and 2 were often playing games in VR. Their age ranged from 19 to 33 (mean: 25, std: 5.2). None of them were wearing glasses and 2 had contact lenses.

4.2 Apparatus

An HTC vive virtual reality setup with an integrated eye tracker from Tobii [18] was used in the study. The eye tracker worked at a frame rate of 120Hz and the average gaze from both eyes was used as the main gaze ray.

4.3 Target size and Menu layout

The main target was a sphere placed in front of the head and 70 cm away from the user. The target size was $10°$ of visual angle. We wanted to set the target large enough so that the inaccuracy of the eye tracker doesn't affect our evaluation particularly when the target is seen from extreme viewing angles. The menu was on a 2D plane and had a circular layout with 8 circles inside. In each trial only one of the widgets were displayed and filled with a color. The menu was displayed at the same distance as the target in the Head conditions, however, the user could hold the menu at a closer distance in the Hand conditions. The menu layout and its appearance was identical in Head and Hand conditions when the user was holding the controller straight at $40cm$ away from his/her head (comfortable distance to hold the controller). The size of the widget was slightly bigger than the target ($15°$) and it was $20°$ away from the center of the target. Figure 3 shows a screenshot of the user's view from our user study.

4.4 Design

The experiment consisted of four conditions: EyeSeeThrough and two-step selection technique, where each technique was tested with hand-held menu and head-attached menu (2 Techniques (EyeSeeThrough vs two-step selection) × 2 Modalities (head vs hand)). Each condition had 80 trials which was divided into two blocks with a rest between blocks. The conditions were counterbalanced across participants and the position of the tool in the menu was pseudo-randomized making sure that each of the 8 positions were taken 5 times.

4.5 Procedure

During experiment, participants were standing and had free space around them to move their hand. The wires attached to the VR device were hanging from the ceiling and we made sure that the participants can rotate their head freely. After completing a demographic questionnaire, the subject was asked to put on the VR headset and adjust the straps for comfort. The experiment was started with a 5-point gaze calibration and after calibration, participants had a brief training session before each condition where they were introduced with the technique used in that condition. Once they were confident with the technique (2-3 trials), the first block began. After the first block, there was a 10 sec break followed by a re-calibration of the tracker. The subject then completed the second block and after that they were asked to take off the HMD and fill out a questionnaire. Each condition lasted about 5 minutes. The same procedure was repeated for each condition. At the end, there was an overall questionnaire which contained questions and comments on their general preference.

In each trail the user had to color a white sphere using a color widget (tool) shown on the menu. In the two-step selection conditions the subject had to look at the tool, press and hold the touchpad button of the hand-held controller, look at the object and release the button. They could also look at the object first, press the button, then look at the tool and release the button. The subject was free to move and position their head or hand as they prefer. In the EyeSeeThrough conditions, the subject had to first align the tool with the object by moving the head (in the Head condition) or the hand (in the Hand condition), and then click the button. The object was colored after the selection was confirmed and the next trial started after a delay of 1 sec. There was also a tick sound as aural feedback whenever the task was completed.

5 RESULTS

We now present the results of the study.

5.1 Task Completion Time

We calculated task completion time in each trial and conducted a two-way repeated-measure ANOVA to determine the effect of Modality and Technique on the selection time. There was a statistically significant interaction between Modality and Technique on selection time $[F(1,1) = 4.9, p = 0.049]$. Therefore, simple main effects were run using one-way repeated-measure ANOVA which showed that the selection time was significantly lower in our technique compared to Two-Step selection technique in both the Hand (0.58 sec vs 0.98 sec) $[F(1,11) = 24.30, p = .000]$ and the Head (0.68 sec vs 0.90 sec) $[F(1,11) = 14.63, p = .003]$ conditions. The effect of Modality was not significant in the two-step selection technique $[F(1,11) = 1.68, p = .22]$, however, the selection time was significantly lower $[F(1,11) = 7.85, p = .017]$ for the hand (0.58 sec) compared to head (0.68 sec) in our technique. which indicates that the users performed the alignment faster using their hand compared to head.

5.2 Questionnaires per Condition

After each condition participants completed a questionnaire and express their perceived easiness, speed, precision, learnability, and physical effort, allowing to better understand how users perceived the interaction.

Five of the participants found the alignment technique tiresome in the Hand condition. This was mainly because of the need for holding the menu up by hand throughout the recording. Three of the participants found the the alignment technique to be tiresome in the
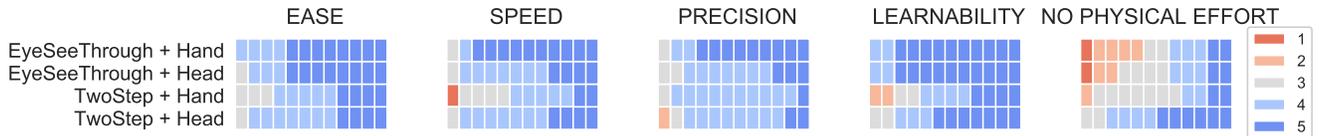
Figure 4: The responses to the questionnaire after each condition (1: strongly disagree ∼ 5: strongly agree).Each square in the plots represents one vote.
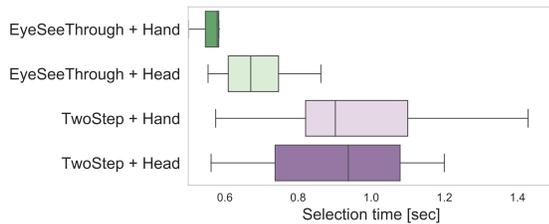

Figure 5: Selection time in different conditions.

Head condition which may indicate that the widgets should not be placed too far in the periphery (18° in our case) in a cockpit style menu. In general, participants had no trouble performing the task in any of the conditions which shows the fairness of the experiment design. They also found the interaction technique fairly easy to learn.

Over 70% of the participants strongly agreed that they were actively using their eye movements and that they were conscious that the technique used their gaze in the two-step selection technique. In contrast, only 25% had this opinion about the EyeSeeThrough technique. Interestingly, the participants were more conscious (more than 20% of them) about the use of their gaze in the when the alignment was done by head compared to hand interface.

5.3 Ranking

After the last round, the participants were also asked to rank different conditions by answering two questions and to give their general comments about each condition. The first question was "Do you prefer two-step selection technique or alignment-based technique?". 9 participants preferred the EyeSeeThrough technique and 3 preferred the two-step technique. We asked the participants about their preferred choice for the modality in our alignment technique: "In which condition was it easier to align the target with the color? Hand or Head". 8 of them preferred the Hand condition and 4 preferred the Head condition. Overall, the EyeSeeThrough + Hand was the the most preferred condition.

5.4 Questionnaire: One- vs. Two-step Interaction

After all conditions, participants expressed their opinion about easiness, speed, precision, and effort for EyeSeeThrough compared to the two-step. For instance, users stated "It felt natural to use alignment" (P4), or, "it was easier to learn and less thinking was required" (P12). Some people also commented about the alignment step: "I liked how there were less steps" (P11) or "The alignment didn't always feel like an explicit step of the interaction. I could focus the gaze on a single target. Which was less straining" (P5).

Gaze Shift Although some participants thought that shifting gaze between target and tool was less tiring than moving the tool (Figure 4), overall, they preferred the EyeSeeThrough technique that requires less gaze shifts. Some people stated this in their comments on the two-step conditions: e.g., "It was a bit annoying to having to constantly move my gaze around [...], I can imagine this working for more complex environments/UIs, but it required the two steps which

felt slower, and required effort to consciously look at and select the colour and then the ball." (P5).

Hand-based EyeSeeThrough Majority of the participants were less positive about the physical effort of the hand-based alignment after trying all the conditions. For example, "for multiple selections it felt very tiring having to move the palette around, or even just holding it any place was quite tiring and uncomfortable" (P1), "The extra hand movement can be very straining. Especially if the targets are at high position where I have to extend it far to be close to a target" (P5). However, they thought it's easier and more precise to perform the alignment with hand compared to head: e.g., "It felt a lot quicker and easier to use, more of a fluid movement" (P4), "It's more convenient, feels more natural... Arm movement is easier than the head movement. " (P9), and "A trade-off between effort required and perception of quicker and free movements." (P6). Some further expressed having full control of the menu as an advantage of the hand-held palette: "I liked the fact that I could position the palette anywhere I wanted with my hands, and in the event the palette was in the way I could move it out the way" (P1).

Head-based EyeSeeThrough Some of the participants stated that aligning with head was less tiring and that it required "less physical movement"(P9) and "least amount of energy" (P1), however, they found it "awkward" (P2), "annoying and disorienting in the long run" (P5), and "unnatural" (P4). They liked the fact that it allows for hands-free interaction, e.g., "For application in AR systems I feel this would be the best approach (plus alignment) as you want to be completely hands free and the alignment based method gives enough precision with less strain."(P3).

Three of the participants commented about the form factor of the cockpit style menu. They thought it give them a faster access to the menu items, however, they found it distracting particularly for long-term use as the menu is always visible and cluttering the view. P1 stated that "Some way of turning the cockpit on and off would be great (if this were to be used in a real-life scenario).".

6 EXAMPLE APPLICATION

To demonstrate the practicality of EyeSeeThrough for interaction in 3D virtual environment, we developed an environment control application where the user can try the EyeSeeThrough technique to interact with a table light, a TV, a color-changing light, and an interactive door in a virtual living room where each object had its own menu.

Table light: We prototyped a table light as an example of a basic interactive object that has a relatively small size which allows the menu items to be presented around it (Figure 6).

Television: The television object had more options available in the menu, and due to its size multiple tools could be overlaid simultaneously. This more shows the main role of gaze in our technique as the gaze implicitly defines the intended tool.

Color-changing standing light: A color changing light showed the feasibility of EyeSeeThrough for controlling two continuous parameters simultaneously, such as the brightness and the color of a light. We also wanted to demonstrate a form of temporary action (e.g., previewing the mode) before the confirmation command is being issued.
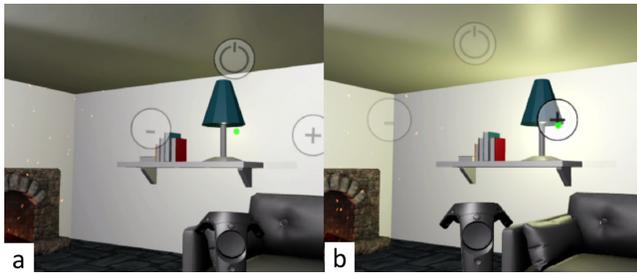
Figure 6: The table light demo with three widgets on its menu for adjusting the light level and turning ON and OFF. (a) The user is holding the menu around the light while looking at the light. (b) The user increases the intensity by holding the right tool aligned with the object. The green dot indicates the user's gaze point.
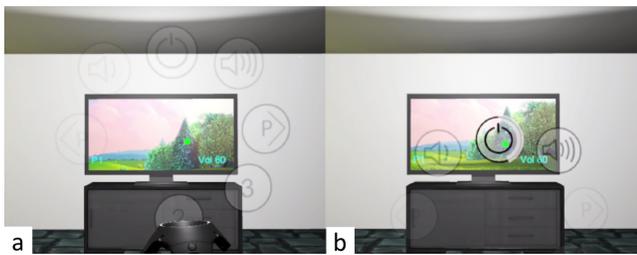

Figure 7: Virtual TV with 8 widgets around it. The right figure shows the user looking through the ON/OFF button to turn the TV off.
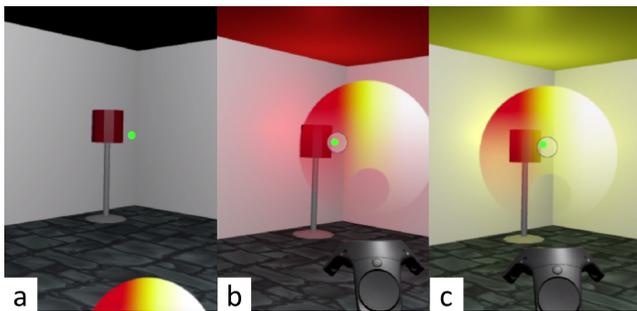

Figure 8: (a) The user is looking at a light while holding the menu in hand. The mode of the light consisting two continuous parameters (color and intensity) can be adjusted depending on where the gaze ray intersects the 2D menu. (b) and (c) show the light when it's been seen through the red and the yellow parts of the menu.
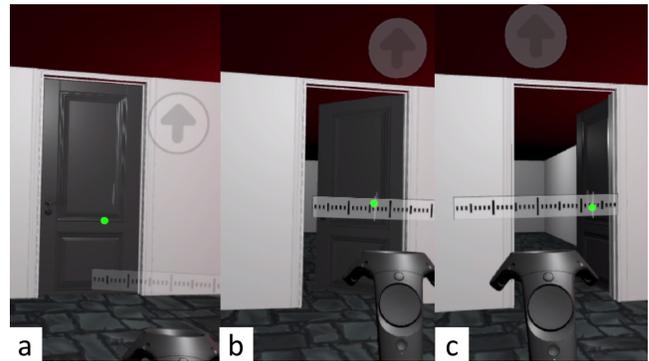

Figure 9: A door as an example object with two kinds of widgets on its menu: a button-like widget to execute the action of entering the door, and a one-dimensional slider for continuously opening and closing the door (e.g., b & c).
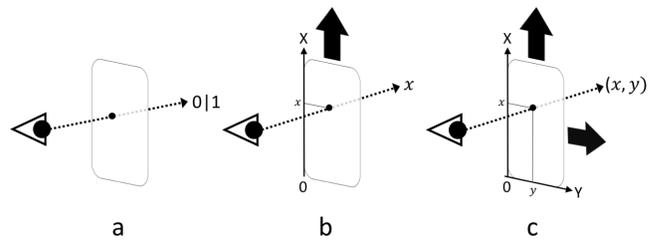

Figure 10: 3 different types of tools implemented in our application. (a) A binary tool, (b) one-dimensional slider, and (c) two-dimensional slider.

Interactive door: With an interactive door, we showed a combination of a binary button that teleports the user to the other room, and a one-dimensional slider for gradually closing and opening the door.

We implemented our technique using two forms of user interfaces: hand-held menu and head-attached menu, and two forms of triggers: Dwell and button press using the hand-held controller. We also implemented different types of tools depending on the type of action they apply to the object:

Binary tool: A simple form of action could be a discrete command to change a binary mode such as 'turn ON' and 'turn OFF'. A binary tool always return a binary value upon confirmation regardless of where on the tool the user is looking through (Figure 10.a). Both the table light and the TV had an ON/OFF tool in their menu. The TV also had three shortcut tools (labeled as 1,2, and 3) that let the user to select a channel. The interactive door also had a binary action tool to teleport the user to the other room. A binary tool could be designed in such a way that the action is continuously triggered (possibly with a predefined time-interval) as long the tool is visually aligned with the object. An example for this were the two volume buttons on the TV menu by which the user could increase or decrease the volume of the TV by holding the volume button aligned with the TV. The system would then sequentially change the volume up or down. The table light also had two buttons for continuously adjusting a the light intensity.

One-dimensional slider: The EyeSeeThrough technique allows for more fine-grained actions, for example, adjusting a continuous parameter (e.g., time or temperature). This can be done by a tool with a slider-like style that returns different values (from a range) depending on where the gaze is intersecting the tool along a specific direction (Figure 10.b). In our application, we implemented an example tool for gradually controlling the opening of our interactive

door (Figure 9.b&c). In order for the user to perform a smooth interaction with these kinds of tools, they should keep the gaze fixed on a specific point of the object while sliding the tool with their head or hand. This is because, changing the fixation point results in sudden changes of the location of the point where the gaze and the tool intersect.

Two-dimensional slider: Two-dimensional sliders allow the user to simultaneously adjust a pair of continuous parameters as the tool is sensitive in both directions. Figure 10.c illustrates the operation of a two-dimensional slider. One example for this is the menu designed for the color-changing light implemented in our application where the user could adjust the color and intensity of the light by moving the tool and looking through any particular area of the menu.

Figure 8 shows the menu layout of the color changing light which was a semi-transparent two-dimensional color gradient tool through which the lamp can be seen. Colors on the lower part of the tool are more transparent than those on the top allowing the user to control the intensity of the intended hue color. There was also a blank circular region at the bottom of the tool which could be used to switch the light OFF. The user could see the changes of the light in realtime whilst they are adjusting the tool. This was done by temporarily changing the mode of the light. The confirmation was done after keeping the menu still for 700 ms. There was also a small circle at the center of the light making the selection easier by visually guiding the user to fixate on it while sliding the tool.

In our application, the unnecessary tools were kept hidden in the menu, e.g., the channel shortcut of the selected channel in the TV menu. To avoid detracting the user from their main task, the menu of the color changing light, and the ON/OFF button of the TV menu go hidden after the user confirms the selection.

7 DESIGN CONSIDERATIONS

Implementing different interactive objects with various sizes, different menu layouts, and at different distances helped us explore some of the design considerations which we discuss in this section.

We present our design discussions into 2 subsections mainly addressing the questions of where the menu should be placed and when it should be activated.

7.1 Menu placement in EyeSeeThrough

One main question that arises when designing menus for different types of objects and at different locations (relative to the user) is that where the menu should be placed? While different modalities could be used to move and align the tool with the object, the menu could be placed at different locations relative to the user and the object. We investigated two forms of menus in our study, one attached to the head and one to the hand. The important design factor in both cases was the depth at which the menu is shown relative to the target. This is particularly important because the interaction relies on the ability to align the target and the tool.

There are two main issues that arise when the menu is placed at a different depth than the target. The first issue is related to accommodation problem when the menu is placed at a different focus distance than the target. As a result, the user has to align a blurry version of the tool with the target which may make it difficult to perceive the content shown on the tool. The second issue is about binocular disparity involved in the visual alignment task performed with binocular vision. However, these two issues are only severe when the menu is placed at a much closer distance than the target (e.g., hand-held palette used to interact with a distant object). None of these two issues were significant in our user study because the menu was placed at the same distance as the target in the head conditions and for the hand conditions the users were holding the controller at a comfortable distance which was again close to the target where the disparity issue was not noticeable considering the large size of the target and the menu.
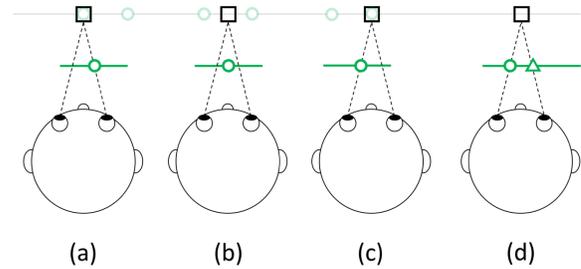

(a)   (b)   (c)   (d)

Figure 11: The figures show different conditions that may occur when aligning a tool widget (circle) with an object (a rectangle). The combined images of the left and the right eye perceived by the person is also shown. (a) The person may align the tool with the line joining the object and the right eye, (c) left eye, or (b) a point between the eyes. (d) Shows how another tool (e.g. the triangle) could accidentally be present in front of the right eye when the person is aligning the circle with the left eye.

7.1.1 Binocular disparity:

Due to binocular disparity, when the person is focusing on the target, they perceive two different images of the tool seen by the left eye and the right eye which makes it difficult to do the alignment. This could affect our interaction technique in two different ways:

User performance: Studies have shown that in a pointing task with hand and two eyes, people normally align their fingertip with a line joining the target and one of their eyes (either dominant or non-dominant) or even a point located between the two eyes [12, 26]. Valkov et al. [41] have looked at the effect of binocular disparity on interaction with 3D stereoscopic displays. They found that the users tend to touch between the left and right images of the target (with an offset towards the target image seen by the dominant eye) when pointing towards a 3D object displayed on a stereoscopic display. Similarly, in our interaction technique, when the task is to align a tool with a distant target, the user may use the left eye (Figure 11.c), the right eye (Figure 11.a) or even both eyes (Figure 11.b). In any case, the users may find the interaction technique unpleasant because it's not clear which image of the tool has to be aligned with the target.

UI: Visual disparity may also set some constraints on the layout of the tools in the menu. Tools may not be placed at certain angles or distances from each other as it may happen that another tool (which is not of interest) is seen over the second image of the object (Figure 11.d).

7.1.2 Alignment detection:

According to the discussion above, in binocular alignment, the tool may intersect with the gaze ray of an individual eye or none of them. Therefore, another consideration is about the way that the system detects the alignment. Using a binocular eye tracker that estimates the gaze for the left and the right eye separately, allows us to cast two gaze rays to the scene one for each eye. The question is, which one of the two rays that intersect the object should be considered as each may pass through different tools. One method could be to monitor both rays to check whether any of them intersect with any tool. However, a more conservative approach would be to cast a set of rays joining the object and any point between the two eyes and checking whether any of the these rays intersect with the tool. In our user study and prototype, we used the combined gaze ray of the left and the right eye.

Another solution could be to have the menu displayed for only one of the eyes (similar to One-Eyed Cursor technique [44]). This makes it easy for the user to perform the alignment and also for the system to detect the alignment using only one eye. The user could
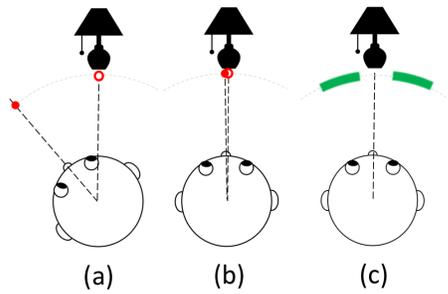
Figure 12: Pre-Alignment step: The head menu in our application was only displayed when the head is aligned with the target. (a) When the user first looks at the object, two small circles appear in the user's view: one attached to the object and the other at the center of the view, (b) the user aligns his head with the target, and (c) the menu appears immediately after the head is aligned with the object.

then align the tool with a line joining the target and the viewing eye, even if it is not the dominant eye [25].

7.1.3 Same depth for target and object

One solution to deal with binocular alignment is to have the tool and the object both at a same depth. A distant menu could still be moved by hand, head or other means. Placing the menu at the same depth as the object could result a smaller looking menu when the object is too far from the user. The menu size could therefore be scaled depending on the depth of the object. Compared to hand-attached menus where the user can easily bring the menu out of their view when it's not needed, in a head-attached menu, the tools are always visible in the view. Seeing the menu in the periphery before the head is fully aligned with the object can be confusing in some applications. Our solution for the head-attached menu in our application demo where there were multiple targets available in the scene was to only activate and display the menu when the head is relatively aligned with the target ($< 2°$). To help the user align their head with the object we used a visual feedback mechanism (referred to as *Pre-Alignment*) explained and illustrated in Figure 12. Two small circles appear in the user's view as soon as the user looks at the object, one attached to the object and the other at the center of the view. The two circles help the user to place the head at the right orientation. The menu appears immediately after the head is aligned with the object.

7.2 Menu activation

Another key design factor that is important is the determination of when the menu should be displayed and activated. In the simplest form, the instantiation step could be taken for granted for example when the menu is available and can be moved by the user all the time. However, having the menu always on especially with the head-attached design, may lead to displaying unnecessary graphical content that clutter the user's view and distract the user from any primary task that they are doing. This type of activation may also not be an appropriate choice when each object has its own customized menu and set of tools. Another solution is to display the menu after the head is fully aligned with the target, as described in Figure 12.

Another design could be that menu is activated by gaze or an extra selection. The menu may appear whenever the user looks at an object or by an explicit request for example when the user performs a particular gesture (e.g. raising the hand or nodding the head). The instantiation could also be handled by a context-aware system that computes users' location and orientation relative to the object and displays menus that correspond to the object (e.g., [3]). Although an extra selection step is needed to instantiate the menu, it's just a one-time step and the menu will remain active afterwards.

8 DISCUSSION

In this paper, we explored EyeSeeThrough, a gaze-based interaction technique that unifies the tool selection and application into one step. This is accomplished by 1) extending the alignment principle of Toolglass from 2D UIs to 3D, and 2) by exploiting eye tracking technology to allow users to see through the tool toward the target. We highlight that eye gaze is used implicitly, as users would naturally look at the right tool and right target they want to manipulate.

We compared our method with a two-step gaze selection technique used as a baseline, with variations of using a cockpit and hand-attached menu. Our result showed that our technique is significantly faster than the two-step selection performed by gaze. The qualitative user feedback also indicates that our technique is easier to perform and easier to learn. Yet, further study is needed to evaluate how the EyeSeeThrough technique compares to more traditional ways to use tools, such as current drawing tools that are based on manual control. Nonetheless, our evaluation provides insights into the novel EyeSeeThrough method and shows promising results.

We further implemented a prototype of a virtual living room scenario to demonstrate the practicality of using EyeSeeThrough for interaction in VR and to explore the design space for EyeSeeThrough. We found that the technique is particularly useful for interacting with context menus that can fade in and out when users attend to interactive objects in the environment. Here, users can simply interact hands-free with a set of commands to the object, providing users shortcuts to the various interfaces in space.

The main concept behind our technique is visual alignment which involves two processes: 1) relative movement of the two aligning items (motion) and 2) eye movements.

Motion: In general, all three interaction components involved in our technique (gaze, object,a and tool) could be moved relative to each other. The concept of the EyeSeeThrough could be generalized such that the aligning items may be anything such as multiple objects in 3D, where the user can convey a relationship and a link between them by aligning one with the other (e.g. defining the position of a piece of jigsaw puzzle by holding it on the hand and aligning it with an area on the puzzle board).

Although our technique could provide hands-free interaction with objects when alignment is done by head, our user study which was performed in VR, showed that people still prefer to use manual input for repositioning the menu in 3D. Future work includes implementing the EyeSeeThrough technique on a see-through HMD in the context of augmented reality allowing the user to interact with real-world objects in a hands-free manner.

Eye movements: Although, with the EyeSeeThrough technique we are substituting a two step-selection task with one step, it is not obvious that the user's gaze remains fixed on a single point during the interaction. Similar to a cursor control task in 2D desktop UIs, people may have different strategies for the eye movements while performing the alignment task. Smith et al. identified three common strategies of eye movement in a pointing task performed by mouse: 1) An object-oriented strategy with the eye primarily on the target (similar to direct hand pointing), 2) an action-oriented strategy with the gaze continuously on the cursor until it is in the vicinity of the target, and 3) a strategy where the users switched their attention back and forth between the cursor and the target [39]. They also claim that the second strategy has never been reported in direct hand aiming tasks in the physical world.

In EyeSeeThrough, the motion is applied to a menu containing a set of tools which is different from the physical hand or a single cursor. The tools are also typically larger than a cursor. We predict that the more familiar the users are with the menu, the more they tend to follow an object-oriented strategy with their focus more on the object and the state of the object rather than the tool. This more follows the two-phase theory of motor control proposed by Woodworth [46] which suggests that the ballistic phase is triggered

open loop behavior and only the second phase near the target is guided closed-loop control behavior. Initial position of the aligning objects in the field of view, their level of importance, the context of the objects, and the modality that is used for moving the tool are some of the other factors that may influence the eye movement pattern during the alignment task.

We compared our technique with two-step selection, however, we argue that these two techniques could nicely be merged allowing the user to interact with an object either by switching their gaze and doing two-step selections, or by performing the selection while looking at the object through the tool.

8.1 Limitations

The EyeSeeThrough is particularly useful when wanting to switch between different tools and targets, but may be subject to limitations in other contexts. This represents the case when users interact with multiple tools at multiple targets (N:N relationship). In other drawing scenarios, users may prefer using traditional tools, when the performance benefit obtained from merging the two steps, is not necessary. For example, when mostly using one tool for multiple targets (1:N), the necessity to have the transparent overlaying menu may occlude content. Similarly, with multiple tools used for one target (N:1). Another limitation may be if there are many tools, which would need the user to perform a lot of alignment for each tool change. To cover these aspects, we aim to explore further extensions of EyeSeeThrough in future.

9 CONCLUSION

In this paper, we introduced EyeSeeThrough as a novel gaze-based interaction technique for VR that fuses two successive selection steps of a action-object selection into one step of alignment. The proposed technique allows the users to keep their visual attention fixed on the interaction object while aligning the menu item with the object. The alignment is based on the explicit movement of the menu relative to the object while implicitly requiring the user to focus their gaze on the object. We have presented a study where we show performance and user preference benefits over a gaze based two-step approach, and our applications demonstrate how the technique can be applied in context of a virtual living room, e.g. to control menus of a TV or lamp from distance, pointing to the possibility of dynamic and rapid interaction with any context menu interface in space around the user.

REFERENCES

[1] F. Argelaguet and C. Andujar. Visual feedback techniques for virtual pointing on stereoscopic displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pp. 163–170. ACM, New York, NY, USA, 2009. doi: 10.1145/1643928.1643966

[2] F. Argelaguet, C. Andujar, and R. Trueba. Overcoming eye-hand visibility mismatch in 3d pointing selection. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 43–46. ACM, New York, NY, USA, 2008. doi: 10.1145/1450579.1450588

[3] S. Benford, H. Schnädelbach, B. Koleva, R. Anastasi, C. Greenhalgh, T. Rodden, J. Green, A. Ghali, T. Pridmore, B. Gaver, et al. Expected, sensed, and desired: A framework for designing sensing-based interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):3–30, 2005.

[4] E. A. Bier, M. C. Stone, K. Fishkin, W. Buxton, and T. Baudel. A taxonomy of see-through tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 358–364. ACM, 1994.

[5] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 73–80. ACM, 1993.

[6] M. Billinghurst, J. Bowskill, N. Dyer, and J. Morphett. An evaluation of wearable information spaces. In *Proceedings of the Virtual Reality*

Annual International Symposium, VRAIS '98, pp. 20–. IEEE Computer Society, Washington, DC, USA, 1998.

[7] D. A. BOWMAN and L. F. HODGES. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages and Computing*, 10(1):37 – 53, 1999.

[8] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[9] D. A. Bowman, C. Wingrave, V. Ly, and J. Campbell. Using pinch gloves for both natural and abstract interaction techniques in virtual environments. In *Proceedings of the HCI International Conference*, pp. 629–633, 2001.

[10] D. A. Bowman and C. A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings IEEE Virtual Reality 2001*, pp. 149–156, March 2001. doi: 10.1109/VR.2001.913781

[11] N. Cournia, J. D. Smith, and A. T. Duchowski. Gaze-vs. hand-based pointing in virtual environments. In *CHI'03 extended abstracts on Human factors in computing systems*, pp. 772–773. ACM, 2003.

[12] J. Crawford, D. Henriques, W. Medendorp, and A. Khan. Ocular kinematics and eye-hand coordination. *Strabismus*, 11(1):33–47, 2003.

[13] P. Dourish. *Where the action is: the foundations of embodied interaction*. MIT press, 2004.

[14] B. M. Ens, R. Finnegan, and P. P. Irani. The personal cockpit: a spatial interface for effective task switching on head-worn displays. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 3171–3180. ACM, 2014.

[15] B. M. Ens, R. Finnegan, and P. P. Irani. The personal cockpit: A spatial interface for effective task switching on head-worn displays. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI '14, pp. 3171–3180. ACM, New York, NY, USA, 2014. doi: 10.1145/2556288.2557058

[16] A. Forsberg, K. Herndon, and R. Zeleznik. Aperture based selection for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pp. 95–96. ACM, 1996.

[17] D. Gerber and D. Bechmann. The spin menu: a menu system for virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pp. 271–272, March 2005. doi: 10.1109/VR.2005.1492790

[18] https://www.tobiipro.com/product-listing/vr integration/. 2018 (accessed Aug 1, 2018).

[19] R. J. Jacob. Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in human-computer interaction*, 4:151–190, 1993.

[20] R. H. Jacoby, M. Ferneau, and J. Humphries. Gestural interaction in a virtual environment. In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pp. 355–364. International Society for Optics and Photonics, 1994.

[21] S. Jalaliniya, D. Mardanbegi, and T. Pederson. Magic pointing for eyewear computers. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, ISWC '15, pp. 155–158. ACM, New York, NY, USA, 2015. doi: 10.1145/2802083.2802094

[22] J. Jankowski and M. Hachet. A Survey of Interaction Techniques for Interactive 3D Environments. In M. Sbert and L. Szirmay-Kalos, eds., *Eurographics 2013 - State of the Art Reports*. The Eurographics Association, 2013. doi: 10.2312/conf/EG2013/stars/065-093

[23] J. Jankowski and M. Hachet. A Survey of Interaction Techniques for Interactive 3D Environments. In *Eurographics 2013 - STAR*. Girona, Spain, May 2013.

[24] P. Kabbash, W. Buxton, and A. Sellen. Two-handed input in a compound task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pp. 417–423. ACM, New York, NY, USA, 1994. doi: 10.1145/191666.191808

[25] A. Khan and J. Crawford. Coordinating one hand with two eyes: gaze-dependent reversal of ocular dominance in a pointing task. In *Soc Neurosci Abstr*, vol. 27, 2001.

[26] A. Z. Khan and J. D. Crawford. Coordinating one hand with two eyes: optimizing for field of view in a pointing task. *Vision research*, 43(4):409–417, 2003.

[27] M. Kytö, B. Ens, T. Piumsomboon, G. A. Lee, and M. Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 81:1–81:14. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173655

[28] J. Liang and M. Green. Jdcad: A highly interactive 3d modeling system. *Computers & graphics*, 18(4):499–506, 1994.

[29] R. W. Lindeman, J. L. Sibert, and J. K. Hahn. Hand-held windows: towards effective 2d interaction in immersive virtual environments. In *Virtual Reality, 1999. Proceedings., IEEE*, pp. 205–212. IEEE, 1999.

[30] D. Mardanbegi, D. W. Hansen, and T. Pederson. Eye-based head gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pp. 139–146. ACM, New York, NY, USA, 2012. doi: 10.1145/2168556.2168578

[31] M. Mine et al. Virtual environment interaction techniques. *UNC Chapel Hill computer science technical report TR95-018*, pp. 507248–2, 1995.

[32] M. R. Mine, F. P. Brooks, Jr., and C. H. Sequin. 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, pp. 19–26. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1997. doi: 10.1145/258734.258747

[33] H. K. Pahud, M. and B. Buxton. In-place: Interacting with large displays. *TechNet Inside Microsoft Research Blog Post*, Oct 4th, 2012.

[34] K. Pfeuffer, B. Mayer, D. Mardanbegi, and H. Gellersen. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, pp. 99–108. ACM, New York, NY, USA, 2017. doi: 10.1145/3131277.3132180

[35] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pp. 39–ff. ACM, 1997.

[36] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 36–39, March 2017. doi: 10.1109/3DUI.2017.7893315

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

[38] L. E. Sibert and R. J. Jacob. Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 281–288. ACM, 2000.

[39] B. A. Smith, J. Ho, W. Ark, and S. Zhai. Hand eye coordination patterns in target selection. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, pp. 117–122. ACM, 2000.

[40] V. Tanriverdi and R. J. Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 265–272. ACM, 2000.

[41] D. Valkov, F. Steinicke, G. Bruder, and K. Hinrichs. 2d touching of 3d stereoscopic objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pp. 1353–1362. ACM, New York, NY, USA, 2011. doi: 10.1145/1978942.1979142

[42] R. Vertegaal. A fitts law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 241–248. ACM, 2008.

[43] J. Viega, M. J. Conway, G. Williams, and R. Pausch. 3d magic lenses. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pp. 51–58. ACM, 1996.

[44] C. Ware and K. Lowther. Selection using a one-eyed cursor in a fish tank vr environment. *ACM Trans. Comput.-Hum. Interact.*, 4(4):309–322, Dec. 1997. doi: 10.1145/267135.267136

[45] G. C. Williams, H. Faste, I. E. McDowall, and M. T. Bolas. Physical presence: Palettes in virtual spaces. In *Electronic Imaging'99*, pp. 374–384. International Society for Optics and Photonics, 1999.

[46] R. S. Woodworth. Accuracy of voluntary movement. *The Psychological Review: Monograph Supplements*, 3(3):i, 1899.

[47] R. C. Zeleznik, A. S. Forsberg, and J. P. Schulze. Look-that-there: Exploiting gaze in virtual reality interactions. *Technical Report CS-05, Tech. Rep.*, 2005.