



FocusFlow: 3D Gaze-Depth Interaction in Virtual Reality Leveraging Active Visual Depth Manipulation

Chenyang Zhang*

zhang414@illinois.edu

University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

Eric Shaffer

shaffer1@illinois.edu

University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

Tiansu Chen*

tiansuc2@illinois.edu

University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

Elahe Soltanaghai

elahe@illinois.edu

University of Illinois at Urbana-Champaign
Urbana, Illinois, USA

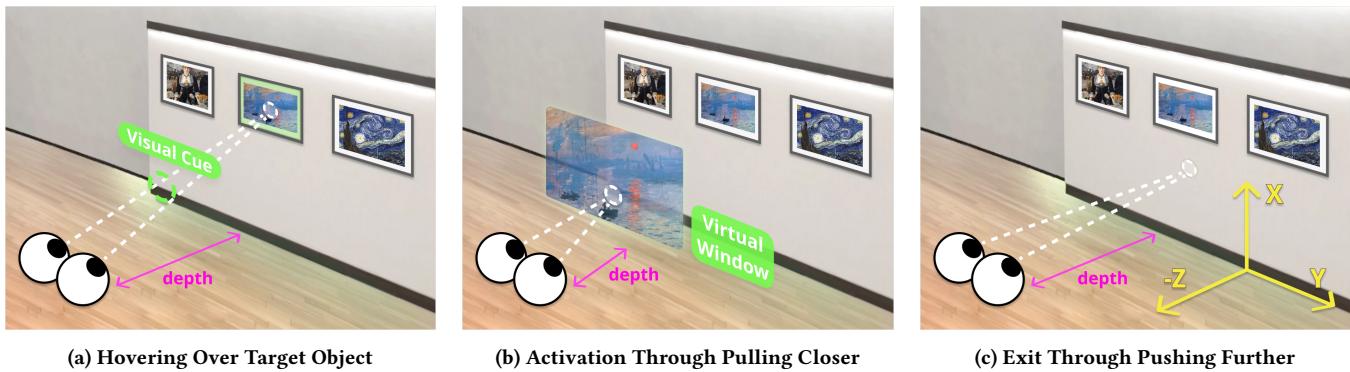


Figure 1: Concept of FocusFlow. (a) When users hover over an object, a visual cue will appear to guide the users to shift their visual depth between layers at different distances. (b) Users can perform hands-free selection to activate a *Virtual Window* in front, by pulling their visual depth closer along the z-dimension. (c) Users can exit the *Virtual Window* by naturally pushing the visual depth further back to the wall.

ABSTRACT

Gaze interaction presents a promising avenue in Virtual Reality (VR) due to its intuitive and efficient user experience. Yet, the depth control inherent in our visual system remains underutilized in current methods. In this study, we introduce FocusFlow, a hands-free interaction method that capitalizes on human visual depth perception within the 3D scenes of Virtual Reality. We first develop a binocular visual depth detection algorithm to understand eye input characteristics. We then propose a layer-based user interface and introduce the concept of “Virtual Window” that offers an intuitive and robust gaze-depth VR interaction, despite the constraints of visual depth accuracy and precision spatially at further distances. Finally, to help novice users actively manipulate their visual depth,

we propose two learning strategies that use different visual cues to help users master visual depth control. Our user studies on 24 participants demonstrate the usability of our proposed virtual window concept as a gaze-depth interaction method. In addition, our findings reveal that the user experience can be enhanced through an effective learning process with adaptive visual cues, helping users to develop muscle memory for this brand-new input mechanism. We conclude the paper by discussing potential future research topics of gaze-depth interaction.

CCS CONCEPTS

- Human-centered computing → User interface design; Interaction techniques; Graphical user interfaces; Virtual reality.

KEYWORDS

Gaze Interaction; Visual Depth; Virtual Reality; 3D User Interface; Eye Tracking

ACM Reference Format:

Chenyang Zhang, Tiansu Chen, Eric Shaffer, and Elahe Soltanaghai. 2024. FocusFlow: 3D Gaze-Depth Interaction in Virtual Reality Leveraging Active Visual Depth Manipulation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*, May 11–16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3613904.3642589>

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. ACM ISBN 979-8-4007-0330-0/24/05
<https://doi.org/10.1145/3613904.3642589>

1 INTRODUCTION

Gaze interaction has gained popularity as an input method for 3D interaction in mixed reality (XR) headsets [2, 19, 38], where the eye movements and gestures would be used as an input. Current gaze-based XR interaction methods often rely on gaze as a means of pointing, supplemented by other inputs such as hand gestures [49], head movement [38], or dwell time [28, 33] for selection confirmation to avoid the so-called "Midas Touch" problem [42] – the problem of distinguishing intentional gaze inputs from involuntary fixations as the users perceive the scene. However, these works primarily utilize the direction of gaze and overlook the valuable *visual depth* information, representing an additional free input dimension along the z-axis (Figure 1). To see how, imagine a **window** through which you can see scenery outside. By bringing your gaze focus closer to the window, you blur out the distant scenery and are able to perceive dust particles on the window surface instead. Another example is autostereograms [41], where slightly adjusting your visual depth allows you to perceive a 3D image (see example in Appendix A). In both of these examples, users actively manipulate their visual depth to select which object to observe. In this research, we aim to leverage this natural eye behavior and convert it into an intuitive and learnable interaction input in Virtual Reality (VR).

More recent works on gaze-based VR/AR interaction demonstrate the great potential of visual depth as an interaction input to solve the Midas touch problem. These methods either guide the user to look at physical or virtual objects at different depths [3, 27, 37, 44, 45] or rely on voluntary eye convergence and divergence [13, 15] by asking the users focusing on the nose or imagining to fixate on some point behind the display plane. These works prove the feasibility of leveraging visual depth as an input for VR headset, however, the usability and learnability of this new interaction design are not fully explored. Failure-prone and difficult-to-learn experience can lead to user frustration and fatigue, making the interaction unusable. How to guide the users to learn to manipulate their visual depth as a reliable input is still a research gap.

In particular, leveraging visual depth as an interaction input is challenging due to several practical factors. First, the natural eye movement is inherently unstable, resulting in noisy visual depth estimates. Moreover, the eye tracking modules in the headsets suffer from limited frame rate and jitters, adding extra noise to depth estimates. So, a usable interaction mechanism is required that is robust against imprecise eye input signal. Second, user's visual depth fixation naturally relies on the presence of objects. So, it may feel counter-intuitive if the user is asked to transition to a depth without any visual referent. So, it is crucial to ensure that novice users can intuitively and effortlessly learn and use this interaction and a carefully designed learning procedure is needed to guide the users on how to master this new interaction mechanism. To tackle these challenges, we employed an empirical research approach. Initially, we devised a visual depth detection method for off-the-shelf VR headsets and conducted experiments to investigate the properties of visual depth input. Building upon the insights gained from these experiments, we subsequently developed our proposed user interface and incorporated a learning procedure with visual cues to help users adapt to the depth-control process.

As such, we present *FocusFlow*, a novel gaze-depth interaction method that realizes the concept of **Virtual Window** (just like the examples mentioned above) by utilizing the 3D nature of immersive scenes and the innate human capability to shift focus across different visual planes. FocusFlow employs the eye trackers in VR headsets to estimate the user's visual depth. It then proposes a *layer-based user interface*, where each layer acts as a transparent Virtual Window positioned at varying depths. These windows become interactive when a user's visual depth aligns with their location, allowing for seamless interaction without visual clutter. Specifically, FocusFlow selects interaction targets based on the user's gaze direction and activates Virtual Windows—initially hidden within the VR scene—through voluntary changes in visual depth (as illustrated in Figure 1). This mechanism facilitates a range of applications, from quick previews and system panel activation to detailed zoom lenses for accessibility, enhancing the VR experience without overwhelming the user interface.

To help users master this new interaction mechanism, we incorporate *visual cues* in the UI design and a learning process that allows users to establish a clear connection between the eye behavior and visual depth control, facilitating an intuitive and effortless interaction for novice users. The key intuition is that users can develop muscle memory for freely shifting their focus between different visual planes even if empty of objects. We propose two learning strategies, *in-stages learning*, and *adaptive learning*. The in-stages learning uses a series of *strong* to *weak* visual cues at a certain depth (e.g. where the transparent virtual window is located) so the user can practice visual depth control by just moving her gaze from the target object to the cue. While these cues are effective in practicing visual depth control, they also cause visual interference for the user during non-interaction times. The proposed *adaptive* visual cues addresses this issue by adaptively changing the transparency of visual cues as the user gets closer to the intended visual depth. The transparency level of the cue also serves as a feedback for the user to better control their visual depth changes. To help master this interaction, we design the adaptive learning process gradually reduces the guided depth range until reaching no visual cue.

To evaluate the usability and learnability of FocusFlow, we conducted quantitative and qualitative evaluations, considering input efficiency, accuracy, and cognitive load. Results from 24 participants indicate that FocusFlow offers an efficient, easy-to-learn experience for hands-free selection tasks, effectively addressing the Midas Touch problem while maintaining natural interaction. Furthermore, our findings suggest that the adaptive visual cues in our design enhance users' depth perception during visual depth transitions. Additionally, the adaptive learning strategy improves users' proficiency with our gaze-depth interaction method.

The contribution of this work comprise:

- (1) We analyze the characteristics of gaze-depth estimates and propose strategies for utilizing this imprecise input information for interaction design.
- (2) We develop a layer-based user interface for gaze-depth interaction by realizing the concept of the "Virtual Window". This interface supports intuitive and robust hands-free selection in VR while avoiding Midas Touch problem.

- (3) We design three visual cues that provide users with depth perception, and two learning strategies, in-stages learning and adaptive learning, to guide novice users in adapting to depth control and developing muscle memory for gaze-depth interaction.
- (4) We conduct a user study and a case study to assess both usability and learnability of our proposed system, and compare the effectiveness for different learning strategies.

2 RELATED WORK

In this section, we first summarize different eye behaviors and the corresponding input techniques which can be used for gaze interactions. Based on this, we then review the evolution of gaze-based interaction in VR and some representative methods to identify the research gap. Lastly, we discussed design space research on user interfaces that integrate gaze input.

2.1 Eye Behaviors and Input Techniques

Eyelid Functions. Eyelid movements, ubiquitous in daily life, also hold value as input mechanisms. **Blinking** and **winking**, rapid opening and closing of the eyelid, serve to distribute tears across the eye's surface. Despite their inherent simplicity, these actions can be random and noisy, complicating the balance between efficiency and accuracy. Consequently, most input techniques leveraging blinking and winking are geared towards accessibility applications [14]. Another area of interest is the detection of **eyelid openness** [4], a measurable value that can be used to control objects [48].

Eye Movements. Eye movements have the potential to serve as input signals, as they can reflect attention direction. Notably, **fixation** is the act of maintaining a steady gaze on a specific location, while **saccade** involves quick, darting eye movements that suddenly shift the fixation point. Feit et al. [8] explored the utilization of saccades and fixations for accurate interaction input, laying the groundwork for many gaze-based interactions. **Smooth pursuit movements** allow our eyes to steadily track a moving object. Inspired by this, researchers have proposed several pursuit-based gaze input methods for tasks such as selection [7] and text entry [1, 50].

Vergence. Vergence is a specific type of eye movement that represents the coordinated movement of both eyes in opposite directions to sustain binocular vision. As a result of such eye movement, the user's visual depth will change. Previous works such as Verge-it [3], employ vergence eye movement as an interaction input to select objects. In these techniques, the user is either asked to follow the motion of an object which results in vergence changes, or to look at their nose to create voluntary vergence control. We build our proposed interaction method upon these eye movements and vergence control systems, but we propose an intuitive user interface design that allows the user to perceive **visual depth** and actively use it for interaction with minimal cognitive load or visual interferences.

2.2 Gaze-Based VR Interaction

Gaze-ray-based Interaction. Eye movements were first utilized for VR pointing tasks by Tanriverdi and Jacob [40], resulting in

faster interaction compared to hand-based gesture methods. Building on this, Sidorakis et al. [39] introduced a VR keyboard input method that employs gaze pointing, leading to fewer typing errors than conventional interfaces. Mardanbegi et al. extended this approach with EyeSeeThrough [24], a VR selection method that aligns the intended object with a tool menu using the user's gaze ray. Piemsomboon et al. [33] further explored the application of gaze direction in VR, proposing different *gaze ray selection* methods that only leverage the direction of gaze. However, a direction-only gaze selection method causes a so-called Midas Touch problem [12], where unintentional inputs might be recognized as commands.

Dwell-time-based Gaze Interaction. Researchers have attempted to mitigate Midas Touch problem by incorporating dwell time [18, 21, 23–25, 32] or similar attention patterns [33]. While these **dwell-based gaze selection** techniques can reduce false triggers, they also slow down the input response due to the added time required for dwell validation, which degrades the interaction efficiency and user experience.

Multi-modal Interaction. Another approach for addressing the Midas Touch problem is to expand the input dimension such as leveraging *multi-modal inputs*, which combines gaze with another input modality. The typical multi-modal inputs combine the gaze direction and controller inputs [24, 35]. Other works combine gaze and head information. For example, Sidenmark et al. proposed Radi-Eye [38], which combines head and eye movement to validate selections. Wei et al. [47] proposed a prediction method for target selection in AR based on the distribution of eye and head endpoints. More recent papers explore the combination of gaze and pinch [31], gaze and hand alignment [22], humming [9], or other hand gestures [5, 11, 19, 20, 49]. In these systems, the pointing information is extracted from eye movement and selection or confirmation of information are done with hand gesture. The gaze and hand multi-modal interactions are also built into commercial products such as Apple Vision Pro. However, involving other modalities inevitably increases the interaction complexity and may not be applicable in certain applications that require hands-free interaction.

An alternative way to expand interaction dimension is to leverage **more eye-input information**. For instance, unconventional blinking [19, 34, 49] is used as a complementary gaze input to direction, but the proposed technique suffers from unintentional triggering. Yi et al. introduced DEEP [48] which leverages eyelid movement as a new input for controlling to allow users to adjust the visual depth in a scene, thus improving the gaze-pointing accuracy when some objects are occluded. However, this approach relies on dwell time, which still suffers from efficiency and usability problems. In this study, we incorporate visual depth as a novel input dimension in gaze interaction to address the Midas Touch problem in gaze selection, while maintaining interaction efficiency.

Gaze-depth-based Interaction. Gaze vergence is an emerging eye-input information that has been used for VR interaction. Compared to multi-modal gaze interaction methods that leverage hand-held controllers or hand gestures, visual depth offers a completely hands-free interaction mechanism. Previous works have shown the feasibility of visual depth estimations in the VR headsets by using binocular disparity and stereoscopic vergence in VR headset [10, 43]. The concept of visual depth as a new interaction input has explored

in the previous works, either by defining a semi-transparent window at a different focal depth [27, 44, 45], tracking voluntary vergence movements [3, 13], or matching the vergence changes with the depth changes of a moving objects [37] in VR. These studies have confirmed the efficacy of the gaze-depth-based interaction method from a technical standpoint. However, the learnability of this interaction poses a significant concern, as most users are not acquainted with gaze-depth input. There is a clear need for specific designs that instruct users on how to actively adjust their visual depth, along with comprehensive user studies to assess the usability and efficiency of this novel interaction mechanism. This paper targets these limitations and proposes a layer-based UI design to leverage active visual depth manipulation despite unstable depth estimates of existing eye tracking systems. In addition, we highlight an integrated learning process that leverages visual cues to help users learn how to voluntarily shift their focal depth. We also perform a user study and a case study to evaluate user experience and mental model in our gaze-depth interaction process.

2.3 Immersive Gaze-based UI Design

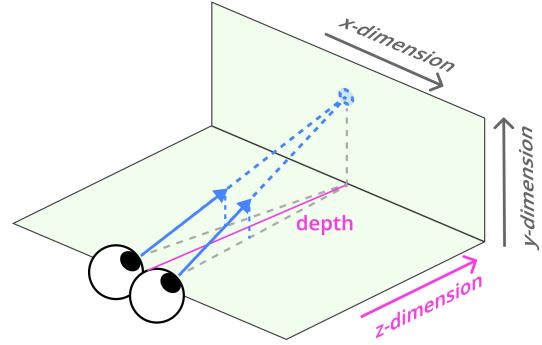
There is a flurry of work on user interface design that incorporate gaze input in VR and AR systems. For example, Pfeuffer et al. [6, 30] proposes a design space for gaze-input based interfaces for AR applications. Hirzle et al. [10] introduced a design space for head mounted devices focusing on human depth perception and technical issues. In this paper, we propose a layer-based UI design for virtual reality that integrates active visual depth manipulation and gaze movement for VR interaction. We evaluate this new interaction and demonstrate the potential of this gaze-only interaction method as an effortless and intuitive method for everyday VR through a learning process.

3 EXAMINING GAZE DEPTH CHARACTERISTICS

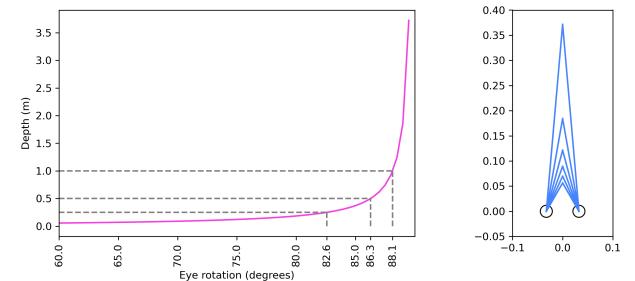
In order to design an intuitive and robust gaze-depth interaction, we first conducted a user study to explore the performance of built-in eye trackers in commodity VR headsets and visual depth characteristics.

3.1 Visual Depth Estimation

In order to calculate the user's visual depth, we use the built-in eye trackers in VR headsets, which capture the positions of eyeballs and converts them into binocular gaze rays originating from eyes and intersecting at a specific depth (see Figure 2a). However, two rays in 3 dimensions may not intersect at a point specially with a small error in gaze direction estimations. To address this issue, we leverage the projection of these two rays onto a plane which ensures the intersection of the two rays. We define the projection plane along the two gaze ray origins (x-axes) and z-axes. As such, the distance between the eye line and the intersection point in this projection plane will be equal to the visual depth. To deal with random eye movements [26, 51] and the resulting depth jitters, we apply a moving average technique with 0.2 seconds of time window to filter out the high frequency depth fluctuations in the depth estimates.



(a) Visual depth calculation. Two gaze rays are projected onto the x-z plane. The visual depth is calculated as the distance between the intersection of projected gaze rays and the midpoint of the user's eyes.



(b) Gaze-depth Sensitivity analysis. For a fixed interpupillary distance, a small change in gaze direction at larger angles (aka larger depths) create much larger changes in visual depth. Therefore, larger visual depths are more prone to noises.

Figure 2: Depth calculation and sensitivity analysis.

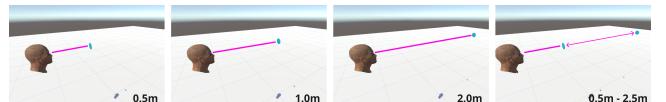


Figure 3: Experiment settings. The participants are required to look at three static objects located at 0.5m, 1.0m, and 2.0m away from themselves. Then they are asked to follow a moving target moving back and forth between 0.5m and 2.5m away with their gaze.

3.2 Experiment Design For Pilot Study

For the apparatus, we use an HTC VIVE Pro Eye headset, which has an FOV of 100° and a 90Hz frame rate. The eye tracking data is recorded by a built-in eye tracker camera at a frame rate of 120Hz and 0.5-1.1° tracking precision. To evaluate the accuracy of visual depth estimates, we invited 7 participants from the campus. For each test, the participant is sitting on a chair while wearing the headset and follows the instructions on looking at certain targets.

We designed two sub-tasks in this experiment. In the first task, participants are asked to stare at a static VR target for five seconds located straight ahead at different distances from the user (0.5m,

1.0m, and 2.0m respectively). The target will be highlighted to capture participants' attention. In the second task, they are asked to look at a moving target that moves back and forth toward the user at depths between 0.5m to 2.5m. The participants are allowed to blink and take a rest if they feel tired in both stages. Figure 3 shows all the tasks in the experiment.

3.3 Takeaways

Figure 4 summarizes the results of the visual depth estimates and visual depth behaviors for the 7 participants. Figure 4a shows the histogram of estimated depths for task 1 with static targets at different depth, represented by different colors. Figure 4b shows snapshots of depth estimates in task 2 (moving target) with and without denoising. We conclude the following visual depth characteristics:

- C.1 **Visual depth is a noisy but usable signal:** In Figure 4a, we can see that estimated visual depths are always presented as a distribution around the target's groundtruth depth. This verifies the unavoidable random eye movements even when looking at a single static object. Nonetheless, we are able to find a relatively clear demarcation between the blue distribution (target at 0.5m distance) and the red distribution (target at 2m distance). This suggests that visual depth can be distinguished at relatively coarse granularity, but it is still a usable signal.
- C.2 **Estimated Visual depth accuracy decreases with the increase of distance:** By comparing the visual depth patterns in Figure 4a and b at different distances, we can conclude that visual depth can be estimated with high accuracy (absolute depth values) and precision (standard deviation of depth distributions) in short distances, but both the depth estimate variations and accuracy drops for larger depths. We then conduct a theoretical sensitivity analysis of the visual depth to the eye rotation and gaze direction. As shown in Figure 2b, the visual depth becomes more sensitive to the eye rotation changes at larger eye rotation values. So, at longer distances (more than 2 meters away), even a small amount of noise or error in gaze direction estimates can result in much larger error in visual depth calculations.
- C.3 **De-noising method plays a critical role.** The raw depth estimation curve (red) in Figure 4b shows persistent fluctuations, especially when the target distance is larger than one meter from the user. After applying our de-noising method, the new depth curve (red) is smoother and drastic depth fluctuations due to random eye movements are eliminated. This indicates that our de-noising method plays a crucial role in pre-processing the input visual depth data.

3.4 Design Rationales Based on Gaze Depth Characteristics

Our pilot study for characterizing the visual depth estimation sets certain trade-offs on leveraging gaze depth as an interaction input. In one hand, gaze depth estimates are mainly reliable in short distances, with larger error in estimated depths as the distance increases. So, for long distance operations, we can only monitor obvious depth shifts which limits the gaze-depth inputs to coarse grained depth changes only. On the other hand, gaze-based interaction is more useful in large virtual environments for interacting

with distant objects, where other modalities (e.g. hand or head gestures) cause user fatigue [49]. So, the main objective of this work is to break this trade-off by taking into account users' spatial perception, input efficiency, and user's learning ability. As such, we define three design goals for FocusFlow. Using FocusFlow, :

- D.1 **Users should be able to achieve hands-free selection efficiently and accurately, without encountering the Midas Touch problem.** The efficiency of a gaze-based interaction is measured by the wait time between the user input and the system response and is expected to be below the user's perceptible limit. The accuracy is indicated by the success rate and false trigger rate of an operation (e.g. selection), which means the selection and deselection intention should be accurately captured by the system.
- D.2 **Users should be able to perceive the depth and accordingly adjust their focal depth for interaction.** Despite the difficulty of users in adjusting their visual depth to a certain length without any reference in space, perceiving the depth by using alternative visual cues can improve voluntary gaze depth changes. The system should offer an intuitive mechanism to grasp the concept of visual depth transitions.
- D.3 **Users should be able to develop muscle memory to provide efficient and effortless interaction.** An immersive virtual scene should be compatible with user instincts to avoid user discomfort [46]. Since voluntary gaze-depth change is a brand-new input method for general VR users, learnability is critical to our interaction approach. Some guidance and feedback should be provided to facilitate the user's learning process.

4 FOCUSFLOW: SYSTEM OVERVIEW

This paper proposes FocusFlow, which consists of three main components to satisfy the design goals mentioned in the previous section:

- **Layer-based UI Design:** FocusFlow presents a layer-based UI design for virtual reality applications that supports gaze-depth interaction. In this design, interactive virtual contents are organized into distinct spatial layers at different visual depths relative to the user. The visibility indexes of these layers are defined based on the user's gaze depth, transitioning between completely opaque when the user's gaze depth matches with the depth of a layer to completely transparent otherwise. Section 5 explains how FocusFlow avoids Midas Touch problem by carefully selecting the layers in the UI and active gaze-depth manipulation to activate a layer.
- **Gaze-depth Interaction Design:** FocusFlow leverages the proposed layer-based UI design and offers a gaze-only interaction mechanism by combining gaze direction and gaze depth. The gaze direction is used for selection of the target object for interaction and gaze depth is used for activating a certain layer in the UI. Section 6.1 discusses a variety of applications that can benefit from this hands-free gaze-only interaction design.
- **Adaptive Interaction Learning Process:** FocusFlow takes advantages of a series of visual cues to help users master this brand-new interaction. The visual cues are designed such

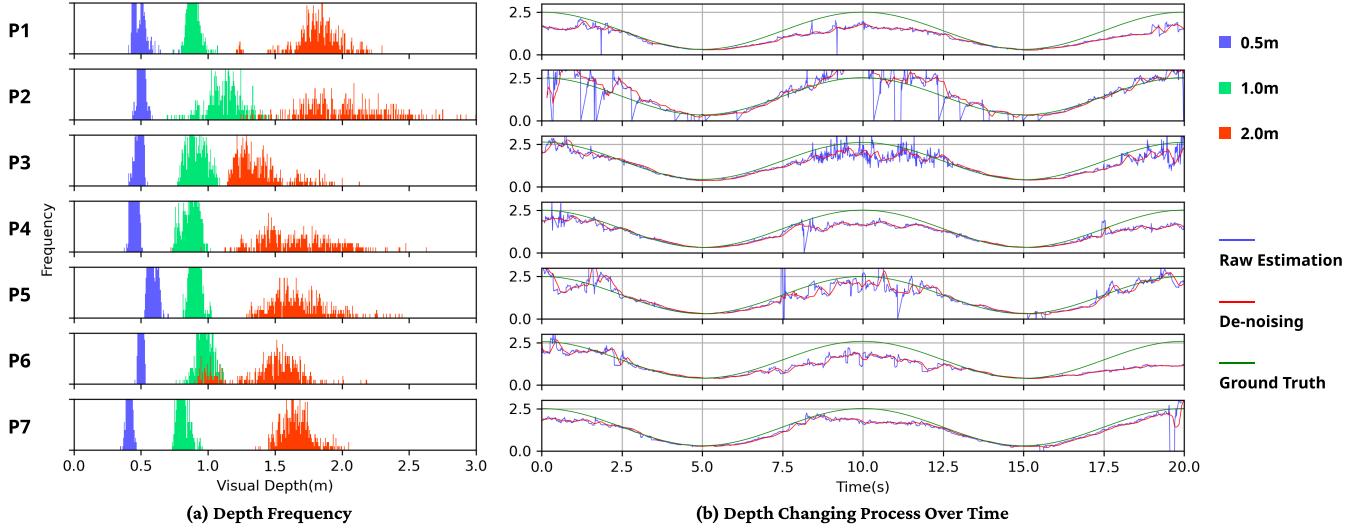


Figure 4: Visual depth detection results. (a) per-participant histogram of estimated depths for three static targets at different distances; (b) snapshot of estimated depth fluctuations for the moving target with and without de-noising.

that they provide users with depth perception and learning opportunities to practice, receive feedback, and master intuitive adjustment of their gaze depth for interaction. Section 7 describes two different learning strategies based on in-stages and adaptive visual cues.

5 FOCUSFLOW: LAYER-BASED UI DESIGN

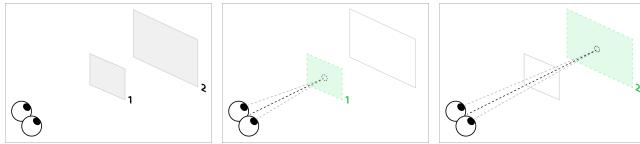


Figure 5: Layer-based UI. The layers are arranged at different depths along the z-direction. The user can match the corresponding layers by changing the visual depth. The matched layers will be activated.

Our vision perception undergoes a significant transformation when there is a change in visual depth. More specifically, a change in the perceived depth causes objects within the original field of view to appear out-of-focus and blurred since they are no longer aligned with our current focal plane. Our goal is to mimic the same intrinsic user experience in the virtual environments by aligning the in-focus objects with the user's perceived depth. To achieve this, we propose a “layer-based” spatial user interface that organizes virtual content into distinct layers. We also propose an activation logic that defines the visibility index of each layer based on the user’s visual depth. These visibility indexes are dynamically adjusted based on shifts in the user’s visual depth. To create an immersive experience, the layer that is matched with user’s visual depth is visible with low transparency so that the user can perceive its position relative to further layers, while the closer layers are completely invisible. As

such, we can enhance the user’s sense of depth perception. Figure 5 shows a high-level overview of the proposed layer-based UI.

5.1 Virtual Window: A Depth-based Selection Widget

Leveraging the proposed layer-based UI, we design *Virtual Window*, a hands-free selection widget with visual depth input. The high level idea is to replicate the user experience of seeing through a “window” or looking at the “window” itself in the virtual environment. Our vision system is capable of ignoring the presence of window and only focus on scenery behind it. At the same time, by bringing your visual focus closer to the window, you blur out the distant scenery and zoom into dust particles on the window surface instead.

Similarly, this virtual Window is an information layer, floating in front of the user, and it is invisible by default. The user can activate the window by bringing their gaze point closer. When the visual depth is matched with the window distance, the information on the Virtual Window will be activated, as shown in Figure 6. This transition process is highly intuitive, as it provides users a sense of “grabbing in” or “taking a closer look” at detailed information when they bring the visual depth closer. Similarly, to send the window to its default transparent status, the user only need to push their visual depth further, back to the portal layer. The window’s visibility change is a response indicating the activation is successful, and also naturally helps users to maintain the visual depth as they are looking at information on the window.

One challenge with this layer-based UI design is potential false triggers or no triggers due to gaze-depth estimation errors. To avoid false triggers, we employ our findings in the pilot study, described in Section 3, that showed a higher depth estimation accuracy at shorter distances. Therefore, we select the virtual window depth within the 2m range of the user at a distance that no virtual object is present. To avoid false negatives and no triggers, we define an *activation zone*, shown in Figure 6, which is a predefined spatial

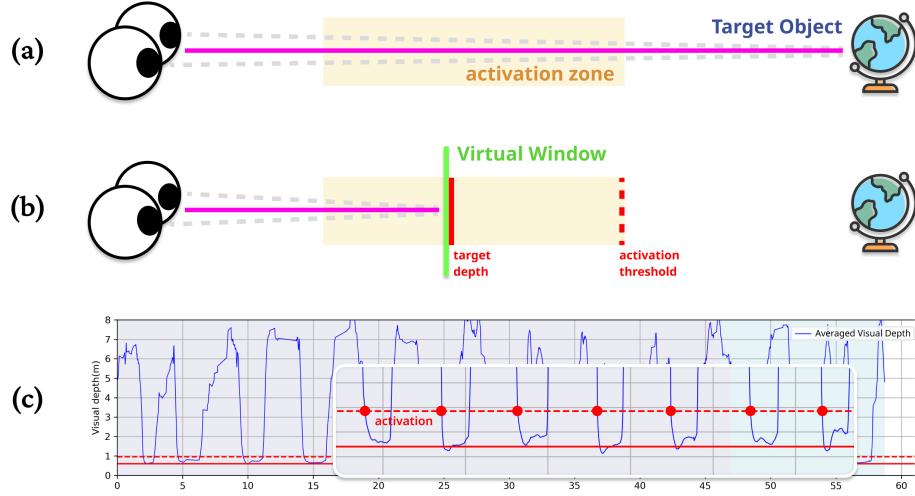


Figure 6: Activation logic. (a) **Pointing:** When the user is looking at the target object in VR environment, the gaze point falls on the object and the visual depth exceeds the activation zone. (b) **Depth Shift:** When the user’s visual depth falls in the activation zone, the Virtual Window will be activated. (c) The gaze depth change in multiple activations.

range around the actual virtual window depth. If the user’s visual depth falls inside this activation zone, the virtual window will be activated.

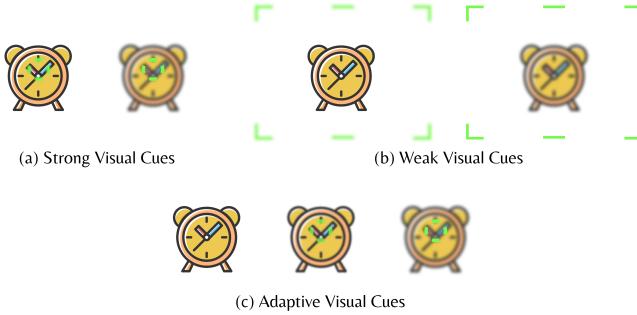


Figure 7: Three different visual cues. (a) **Strong visual cues** appear at the center of the view with constant low transparency. (b) **Weak visual cues** appear at the margin of the view with constant low transparency. (c) **Adaptive visual cues** will appear at the center of the view with variable transparency which depends on the user’s gaze depth.

5.2 Visual Cues

While visual depth transition comes naturally to our vision system, it highly relies on the presence of physical/virtual content at that particular depth. So, it may be at first counter-intuitive for the users to perceive the depth and switch their gaze to a point in the space with no visual feature. Previous works on gaze-depth interaction address this issue by asking users to look at their nose to create vergence [3, 13]. However, this may result in user discomfort and fatigue. Instead, FocusFlow defines three types of **visual cues** that offer intuitive interaction with minimal visual interference.

The first design is a *strong guidance* (Figure 7a), which is a static green circle in the center of the user’s view at the close depth. With this, users can transition their visual depth from far to nearby by directly looking at the circle. However, since the circle is always in the user’s field of view, it may be a strong visual interference for some users. The second design is a frame layout in the corners of the view, called *weak guidance* (Figure 7b). Users can perceive the layer depth through their peripheral vision to transition their visual depth. As such, the strong guidance is a direct referent for the gaze point to focus, but the weak guidance is an indirect referent to the layer depth. The third design is an *adaptive guidance* (Figure 7c), which is also a circle in the center, but its visibility actively changes based on the user’s visual depth. We map its opacity to a certain range of the user’s visual depth. When users are looking at distant objects, the circle is not visible so there is no visual interference. As users look closer, its opacity increases so users can find the circle and manipulate their gaze depth easier. In this adaptive process, the visual cue is not only a depth referent but also an indicator of the user’s visual depth, which could serve as a feedback to the user; from the opacity or transparency of the circle: the closer the visual depth, the more obvious the circle will be. This helps users understand the gaze-depth interaction mechanism and better perceive the depth information in virtual scenes.

6 FOCUSFLOW: INTERACTION DESIGN

Figure 8 illustrates the overall control logic of FocusFlow’s interaction design. The process is initiated with the eye tracker module in the headsets, which continuously monitors the user’s binocular gaze directions. From the gaze direction information, it can then compute the gaze depth to determine the exact point the user is looking at (as described in Section 3.1). The Eye Tracker also employs a ‘moving average’ de-noising technique to refine the gaze data, ensuring that the data is smooth and free from random fluctuations that could lead to depth estimation errors.

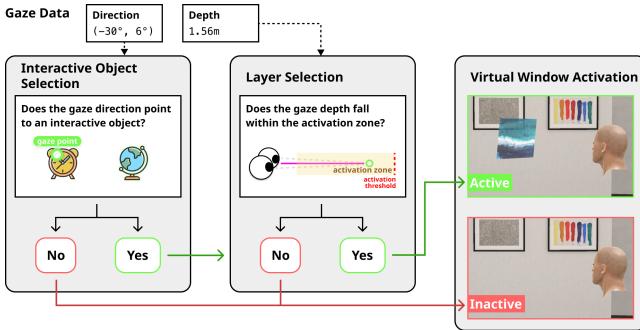


Figure 8: System pipeline of FocusFlow. Eye Tracker provides gaze data to the *Interactive Object Selection* and *Layer Selection* modules, and these two modules then send signals to control the activation of the *Virtual Window*.

The gaze data is then passed to the *Interactive Object Selection* module, which determines whether the user's gaze is aligned with an interactive object in the environment. The module mainly searches for intersection of the gaze rays with individual objects in the virtual scene and moves forward to layer selection module if the an object is selected.

Upon selection of the interactive object, FocusFlow continuously tracks the gaze depth estimates and evaluates whether the visual depth coincides with the activation zone, which is a predefined spatial area where user interaction is intended to occur. If the visual depth falls outside this zone, it is interpreted as a non-interactive state. However, if the gaze depth is within the zone, indicated by a "Yes" decision, it triggers the interaction signal.

Upon satisfying both the direction and depth conditions, the interaction signal is then conveyed to the *Virtual Window Activation* module, which is responsible for managing the active window elements on the user interface. It reacts to the interaction signal by activating or deactivating window elements. For instance, if the user's gaze is directed at and focused on an interactive object within the activation zone, the corresponding virtual window element becomes active with full visibility. Conversely, if the user's gaze shifts away, the virtual window will disappear and return to an 'Inactive' state, as depicted in Figure 8.

6.1 Interaction Applications

The proposed layer-based UI and interaction offers a flexible design that can support different functionalities for the Virtual Windows. We implement and demonstrate FocusFlow functionality across several example applications:

Quick Preview. Given the expected short depth transition delays in our visual systems (i.e. 160-200 milliseconds [16]), users can activate the Virtual Window very fast, making it suitable for quick information preview. As Figure 9a shows, we can display preview information about the selected item on the virtual window, which is very common in scenarios like gaming, training, 3D creation, virtual tours, etc.

Safe Activation. Compared to actions such as gaze direction change, blinking, and finger movement, wide range visual depth shift is a

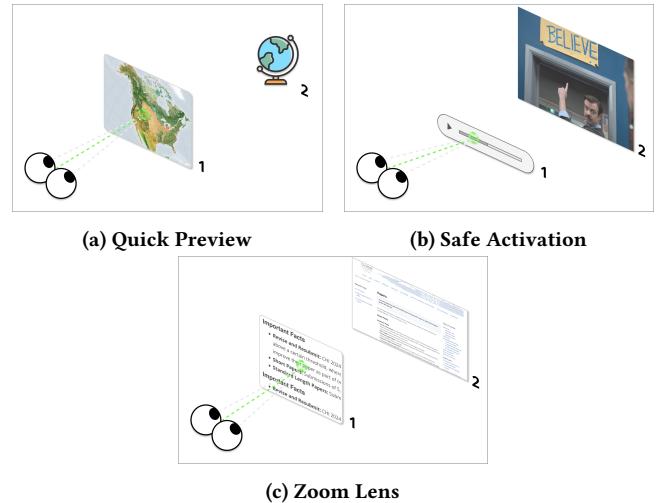


Figure 9: Example applications of Virtual Window.

relatively low-frequency and low-randomness action. Therefore, we can apply the depth activation method to scenarios where we don't want false touches to occur, which we call "safe activation". For example, when browsing a web page or watching a video in full screen, we can activate a toolbar by shifting the visual depth. Since our visual depth is relatively stable, we can avoid false touches that could interfere with our viewing experience. (Figure 9b)

Zoom Lens. Visual depth is an input dimension along z-axis, where occlusion happens. This constraint actually offers logic connection between different layers, for example, the Virtual Window can also be used as zoom lens to magnify the object behind it. (Figure 9c) This design connects the information layer (Virtual Window) and portal layer (environment), i.e. providing a zoomed view.

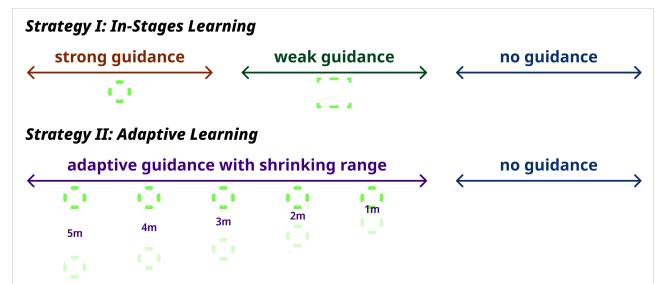


Figure 10: Learning Strategies. In-Stages learning begins with strong guidance at the center of the view, then transfers to weak guidance at the edge, and finally moves to no guidance. Adaptive learning starts with adaptive guidance at the center of the view with variable transparency. As the user progresses, the adaptive range gradually decreases, eventually leading to no guidance.

7 FOCUSFLOW: INTERACTION LEARNING PROCESS

Section 5.2 defines several visual cues that could help the users to intuitively shift their gaze depth. However, we argue that depth change is a learnable input method and can become a muscle memory with a proper learning strategy without the need for strong or even any visual cues. To help users develop this new muscle memory, we propose two learning processes based on different series of visual cues: (1) **in-stages learning** (Figure 10-I), which walks the user through a sequence of strong to weak visual cues, gradually weakening the cue to remove the user reliance on visual cues for gaze depth transitions. (2) **adaptive learning** (Figure 10-II), which leverages the adaptive visual cue with different transparency indexes for the circular visual cue based on the user's visual depth. We describe the technical details of these two learning strategies in the next sections.

In-stages Learning During in-stages learning process, the UI walks the user through a sequence of gaze-depth interactions with strong, then weak, and finally no visual cues. The learning process mainly relies on repetition and gradually weakening the cues in a plane dimension. In addition a fixed learning sequence is expected to be used for different users. This learning process is simple and efficient but lacks feedback to users on their depth control and perception quality.

Adaptive Learning The second learning strategy uses the adaptive guidance. At first, a depth range will be defined for activating the visual cue with different transparency indexes assigned to the visual cue based on the user's visual depth. It should be noted that the visual cue's depth range is much larger than the activation zone for virtual window activation. While the adaptive transparency of the visual cue serves as a feedback to the user, we further expand this feedback-based learning process by gradually reducing the depth range of the visual cue until we reach the depth at which the Virtual Window is located. In addition to feedback offering, this learning process can be flexibly modified for different users by changing the rate of visual cue's depth range.

8 USER STUDY

In this section, we focus on two questions: First, can a gaze-depth input method be intuitively learned to the extent of developing a muscle memory? Second, how efficient is our method and what are the advantages compared with the non-depth hands-free methods? To prove the learnability and efficiency of FocusFlow, we conducted a user study with 24 participants, analyzed the effects of two learning strategies, and compared the performance with the baseline dwell-based selection method as detailed below.

8.1 Participants and Apparatus

We recruited two groups of participants from the campus with 12 people for each group (Group 1 and Group 2). All of these 24 individuals only have occasional or no VR experience. Out of these 24 individuals, 6 of them reported slight shortsightedness (3 in Group 1 and 3 in Group 2), while 18 of them had a standard uncorrected vision. All the participants did not wear glasses during the test. In addition, we invited 2 expert users (E1, E2) who had long-term

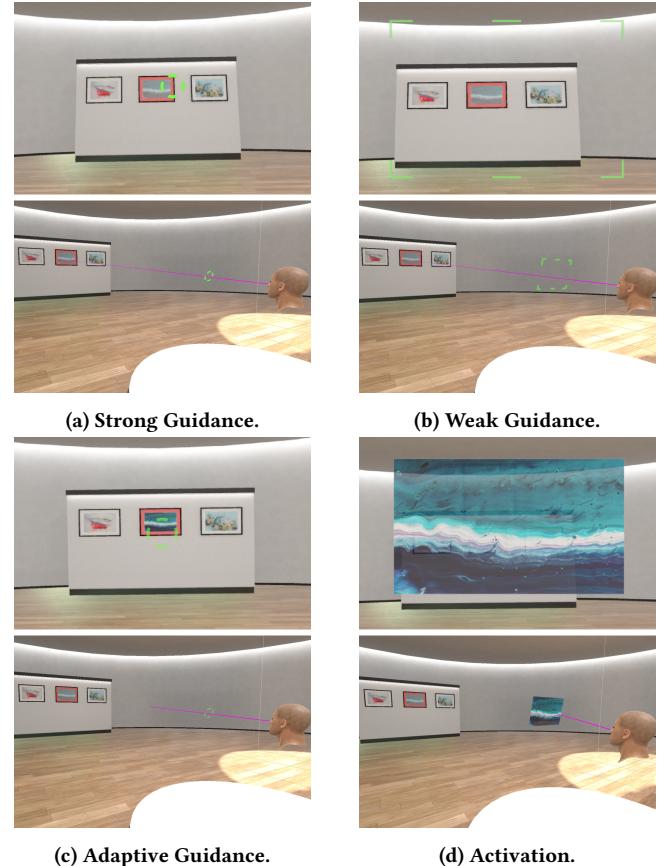


Figure 11: User study in a gallery scenario.

participation in this project as a reference result. The expert users finished all the studies for Group 1 and Group 2. We used the same apparatus as in study 3.2.

8.2 Procedure

Participants were seated during the experiment, and they could take a rest at any time during the entire process. At first, they put on the headset and performed an eye tracker calibration program. Then they were asked to **learn** to do gaze selection tasks in a VR gallery shown in Figure 11, which was to select the target painting to the Virtual Window using the gaze-depth interaction method. This learning process is analyzed in Section 8.3.

Then they put on the headset again to **test** their learning effectiveness and proficiency in operation. Participants were first required to finish 10 selection operations within 5 seconds in the same scene, with the help of visual cues. Then they were required to finish another 10 selection operations within 5 seconds without any visual cues. Once they finished the testing stage, the participants will remove the headset and answer the NASA TLX Workload questionnaires to measure their cognitive load.

Lastly, all the participants in both groups were asked to test the baseline method. They will complete two rounds of 10 selection tasks using the dwell-based gaze interaction method, with time

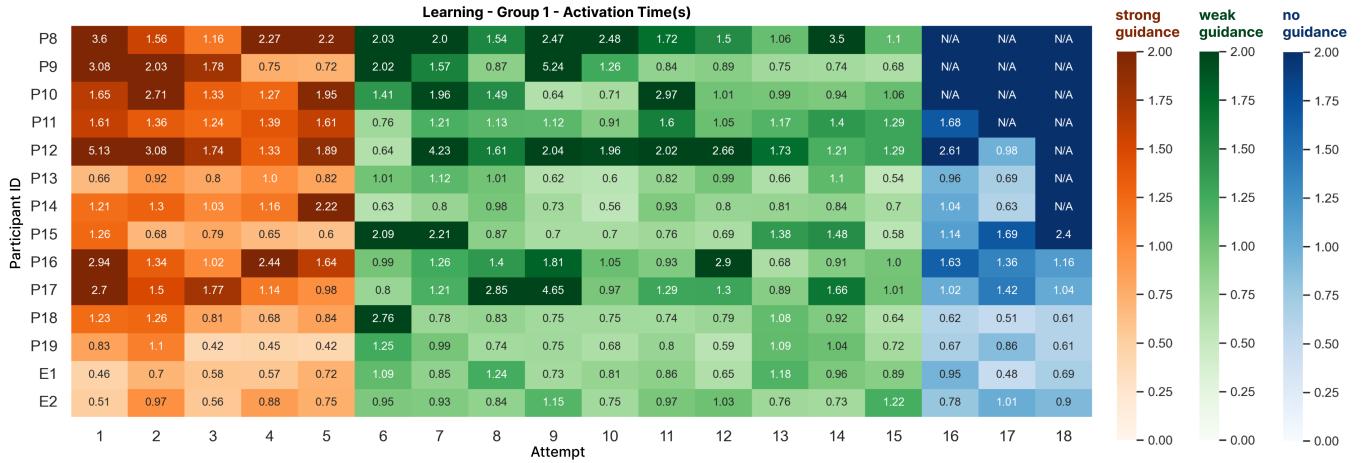


Figure 12: Learning attempts of Group 1. Group 1 adopts the in-stages learning strategy with 5 strong-guidance attempts, 10 weak-guidance attempts and 3 no-guidance attempts.

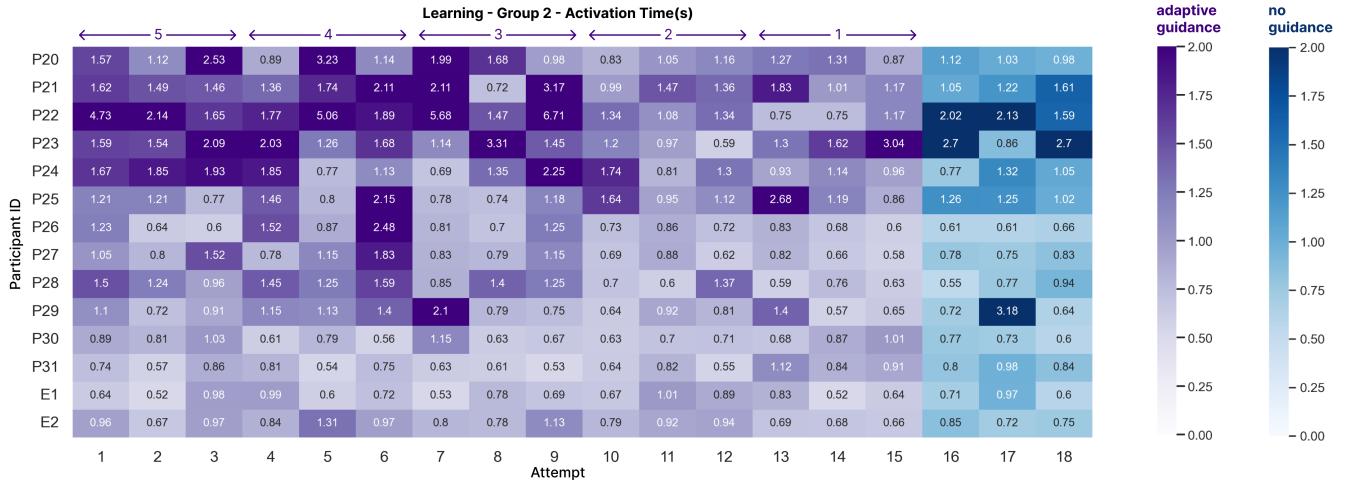


Figure 13: Learning attempts of Group 2. Group 2 adopts the adaptive learning strategy with 15 adaptive-guidance attempts and 3 no-guidance attempts. Within the 15 adaptive-guidance attempts, the adaptive range will shrink for every 3 attempts, starting with 5 meters and ending with 1 meter. The target object for selection is 8 meters away from the user.

thresholds of 0.5 second and 1 second. Finally, all the participants removed the headset and answered a 7-point Likert scale questionnaire about their user experience. The whole experiment took 20-30 minutes to complete.

8.3 Learning Strategy Analysis

In this section, we discuss the detailed performance of the two proposed two learning procedures and their learning capabilities by comparing the activation time, users' gaze depth behavior, and the workload during the learning process. With better learning performance and user experience, the adaptive learning strategy is shown as a more effective way for learning gaze-depth interaction than the in-stages learning strategy.

8.3.1 Learning Procedure. Participants in Group 1 are asked to go through the in-stages learning process. As Figure 12 shows, they first do 5 selections with strong guidance, followed by 10 selection attempts with weak guidance, and finally 3 attempts without any visual cues. On the other hand, participants in Group 2 are asked to go through the adaptive learning process. As shown in Figure 13, they perform 15 selection attempts using the adaptive visual cue. The depth range of adaptive guidance start from 5 meters, shrinks 1 meter every 3 attempts, and ends at 1 meter visual depth range which overlaps with the activation zone of Virtual Window. After this adaptive process, the users are asked to perform selections without any visual cue.

8.3.2 Activation Time. We visualize the activation time of all the participant attempts from both groups in the learning stage in

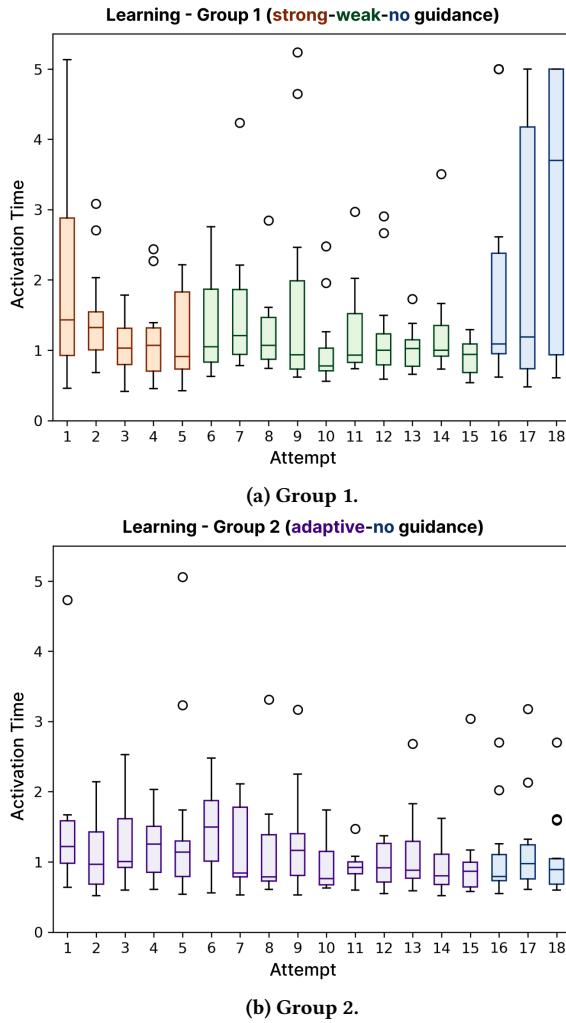


Figure 14: Activation time statistics of all the learning attempts for two groups. the central line in the box indicates the median activation time, while the edges of the box show the interquartile range (IQR), which represents the middle 50% of the data. Group 1's activation times show a wider range with more variability and outliers, suggesting less consistent performance across attempts. Group 2 displays a more consistent activation time across attempts with fewer outliers, indicating a more uniform performance.

Figure 12 and 13, where the color intensity encodes the activation time. Both figures show a trend of decreasing activation time as learning progresses, demonstrated in lighter colors from left to right for each line/participant. We can also see that 7 participants from Group 1 cannot successfully activate the Virtual Window under no guidance setting, while all the participants from Group 2 can successfully do this. This indicates that learning strategy 2 is more capable of helping users form muscle memory during the learning process.

We further analyze the effectiveness of learning by comparing the activation time changes in a statistical way. The box plot in Figure 14 shows the distribution of activation times for each round of attempt within two groups. We can see that the activation time of Group 1 (Figure 14a) shows a decreasing trend in median activation time specifically under the strong and weak guidance setting, but this trend does not extend to the no guidance setting. In comparison, Group 2 (Figure 14b) shows a consistent decreasing trend under the adaptive guidance and no guidance setting, proving its superiority over learning strategy 1.

8.3.3 Gaze Depth Behavior. We recorded the gaze depth data during the learning process to investigate the effect of visual cues on the interaction behavior. Figure 15 shows the raw gaze depth values of two representative subjects (P8 and P26).

P8 adopted the *in-stages learning strategy* and experienced strong guidance, weak guidance, and no guidance activation successively. As Figure 15-A shows, the gaze depth transition of P8 is smooth under the strong guidance and weak guidance setting. However, when entering the no guidance setting from weak guidance (Figure 15-B), P8 made several attempts but can not effectively bring his gaze depth close enough to reach the activation threshold. The reason is it's hard to adjust the gaze depth accurately without any depth referent or muscle memory. The *in-stages learning strategy* fails to help P8 to develop the muscle memory, so as the system switched to the no guidance setting and there was nothing to be a referent, P8 can not shift his gaze depth to the target distance.

The *adaptive learning strategy* shows a better effectiveness on the transition from adaptive guidance to no guidance. As Figure 15-C shows, P26 can still stably shift the gaze depth between close distance and far distance, proving the feasibility of forming a muscle memory. This is because the shrinking assisted range is continuously reduce users' dependence on visual cues, and the changing opacity provides a clear feedback to users to help them understand their input status.

8.3.4 Workload. We employed the NASA Task Load Index (NASA-TLX) to assess the cognitive and physical workload of two learning processes. The result in Figure 16 shows that the no guidance method imposes higher cognitive and physical demands on the users than the guided methods, indicating the muscle memory development requires some effort. In addition, the cognitive load scores of adaptive guidance is slightly higher than the weak guidance method, because its changing capacity brings more information to users to process. However, it also shows a lower performance load score (lower is better), which means eventually the adaptive guidance can offer better activation performance.

8.4 Performance Analysis

The analysis in Section 8.3 revealed that the adaptive learning strategy enhances participants' proficiency with the gaze-depth interaction method in FocusFlow. Building on this insight, this section delves into the performance of Group 2 during their test stages, comparing it to the baseline dwell method to assess the efficiency and accuracy of FocusFlow.

8.4.1 Activation Time. We define 'activation time' as the duration from the moment a user's gaze targets an object to when the virtual

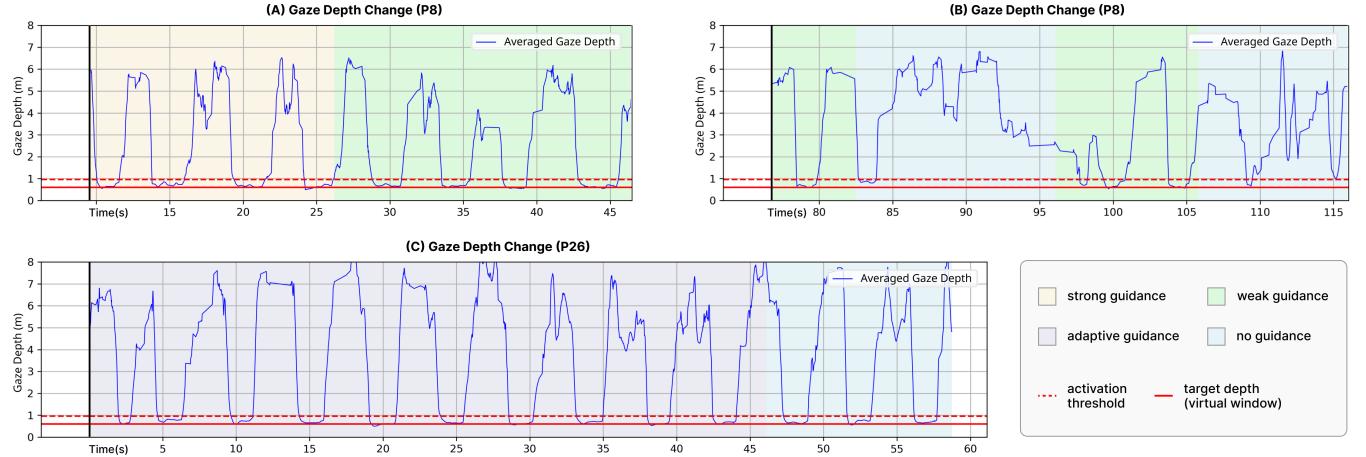


Figure 15: Visual Depth Change. The gaze depth change is visualized as the blue line, and the background color indicates different visual cues settings.

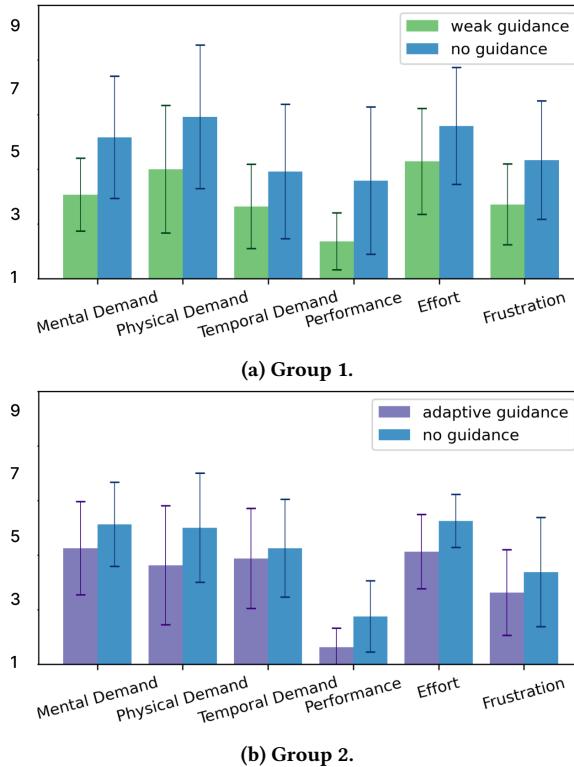


Figure 16: NASA-TLX Results. We measure the subjective workload during two learning process (in-stages learning and adaptive learning) using the NASA-TLX Index. For each entry, lower score means lower workload and better experience.

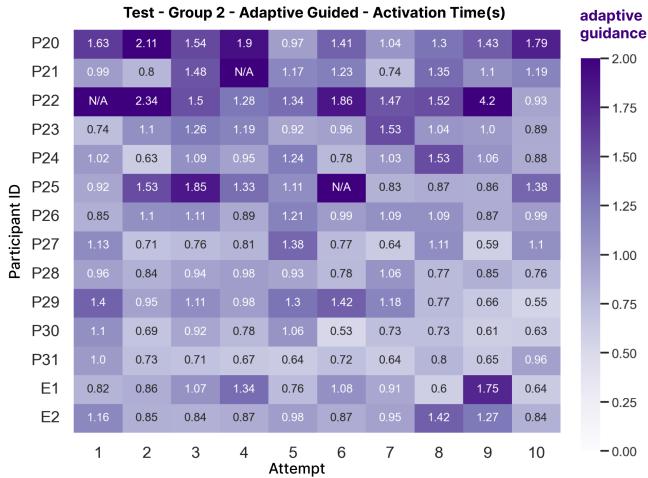
window appears. Figure 17a and 17b illustrate the activation times across all selection attempts by Group 2 during two tests in the testing stage. It's important to note that dark squares marked as

'N/A' represent instances where participants didn't complete the selection task within the 5-second limit. Remarkably, participants in Group 2 successfully completed the selection tasks in the no-guidance test setting. This achievement underscores their high proficiency with the gaze-depth interaction method, particularly when contrasted with the last several attempts shown in Figure 12 for the in-stage learning strategy. Additionally, their activation times in the no-guidance test are comparable to the time in the adaptive-guidance test, suggesting that Group 2 participants have effectively mastered the gaze-depth interaction method without any visual guidance. Moreover, we also compare the activation time between the gaze-depth method (FocusFlow) and the baseline dwell method in Figure 18a. We could find that the two settings in FocusFlow achieve a similar activation time of around 1.3s, which is shorter than the 2-second dwell method, while the half-second dwell method achieved the shortest time.

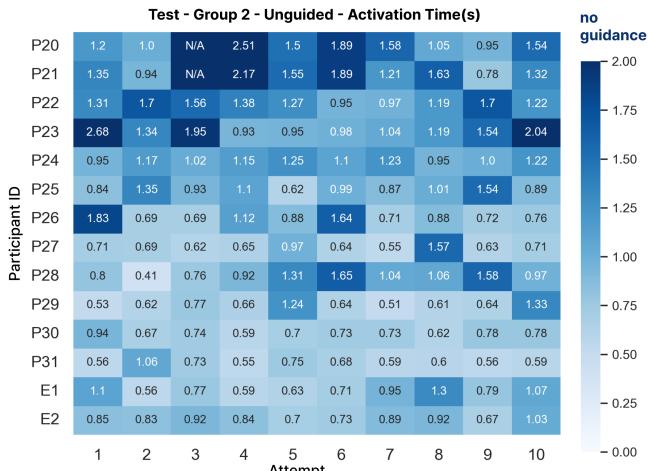
8.4.2 Failure Rate. We define 'failure rate' as the ratio of successful selection attempts to the total number of attempts in each stage, considering an attempt a failure if it's not completed within the time limit. Figure 18b reveals that all four settings exhibit a failure rate below 5%, which suggests that the reliability of FocusFlow is comparable to that of the traditional dwell method.

8.4.3 False Trigger Rate. We define the 'false trigger rate' as the proportion of incorrect selections relative to the total number of selections, with 'incorrect selections' being those where the wrong targets are chosen. As illustrated in Figure 18c, the half-second dwell method exhibits the highest false trigger rate. This may be attributed to the method's overly sensitive nature, leading to an increased likelihood of unintentional target activations. In contrast, both settings in FocusFlow demonstrate a considerably lower false trigger rate, approximately 5%. This rate is marginally higher than that observed in the 2-second dwell method.

8.4.4 Overall Performance. In the preceding paragraphs, we evaluated various metrics of FocusFlow, comparing them against two



(a) Test with adaptive guidance of Group 2.



(b) Test with no guidance of Group 2.

Figure 17: Activation time for the tests of Group 2 in two guidance settings. In the test with no guidance, users in Group 2 can still finish most of the selection tasks in a similar activation time to the time used in the test with adaptive guidance.

baseline dwell time settings as depicted in Figure 18. Overall, FocusFlow demonstrates commendable performance with an average activation time of 1.3 seconds, a failure rate under 5%, and a relatively low false trigger rate of approximately 5%. While the half-second dwell method offers a shorter average activation time and a comparably low failure rate, its high incidence of false triggers could negatively impact user experience. Conversely, the 2-second dwell method, despite its stability and low rates of failure and false triggers, may lead to user frustration due to longer activation times, especially in certain scenarios. Additionally, the dwell method restricts users from continuously observing the target object, which might detract from the overall user experience. Thus, FocusFlow is a reliable and efficient interaction choice with an appropriate

learning strategy applied to enhance the user experience in the VR space.

8.4.5 User Experience. Besides the quantitative metrics we discussed above, we also designed a Likert questionnaire to evaluate the user experience of FocusFlow, focusing on its usability, efficiency, comfort, overall preference, and potential applicability in various VR contexts, and compared the results with the dwell-based baseline method. The result in Figure 19 shows that FocusFlow receives positive feedback from participants, which proves its usability and learnability. Compared to the baseline method, FocusFlow is overall better and obviously outperforms it in terms of efficiency (Q.4, Q.5, Q.6) and overall experience (Q.9, Q.10). It also shows that the eye fatigue when using FocusFlow is more serious than the baseline dwell-based method, which should be an issue that needs to be emphasized and improved upon. In addition, users are wary of using hands-free gaze interaction as an alternative to other VR interaction methods. This suggests that gaze-only interaction cannot cope with all VR interaction scenarios, so compatibility and extendibility with other interaction methods should also be an important design consideration.

9 CASE STUDY

FocusFlow facilitates intuitive and accurate hands-free selections in VR. To demonstrate how users interact with FocusFlow and their perceptions of the process, we present a case study set in a virtual fashion store shopping environment that mimics real-world scenarios. (Figure 20a)

In this environment, each product can trigger a preview window that showcases its profile. This is consistent with the Virtual Window setting encountered during user training. Items within the store vary in distance from the user, ranging between 2 to 7 meters. Users are encouraged to freely explore these products, mirroring a genuine shopping experience. The dwell method and our depth-based method are employed successively to ascertain the experiential difference conferred by the addition of a depth input dimension.

Utilizing **FocusFlow**, users reported a seamless and efficient experience. Initially, they would pan their heads around to acquaint themselves with the virtual space. Gradually, their attention would focus on specific items of interest. At this juncture, visual cues become evident (Figure 20b). “*The appearance of adaptive visual cues hinted at the target depth and gave me some feedback on my gaze depth, allowing me to confidently engage in depth adjustments to activate the virtual window.*” “*I didn’t experience significant visual disruption since the depth of the adaptive visual cues didn’t match the depth of the object I was observing. Thus, I could easily overlook its presence when focused on an item.*” (P19) To access detailed information about an item, users adjusted their gaze to the depth of the virtual window (Figure 20c). This action promptly displayed the product details. Subsequently, by shifting their focus, users could effortlessly continue to survey the environment. “*The activation and deactivation mechanism was very fluid. It felt as though a virtual window was consistently present; and whenever I looked at it, information would display.*” (P8)

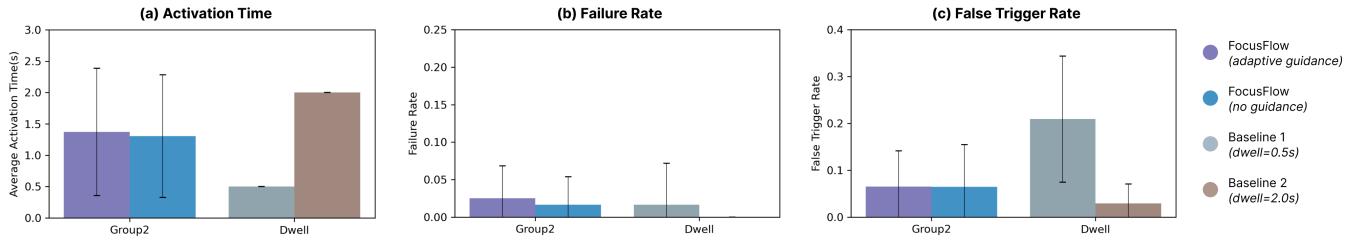


Figure 18: Performance comparison between FocusFlow and dwell methods in different settings.

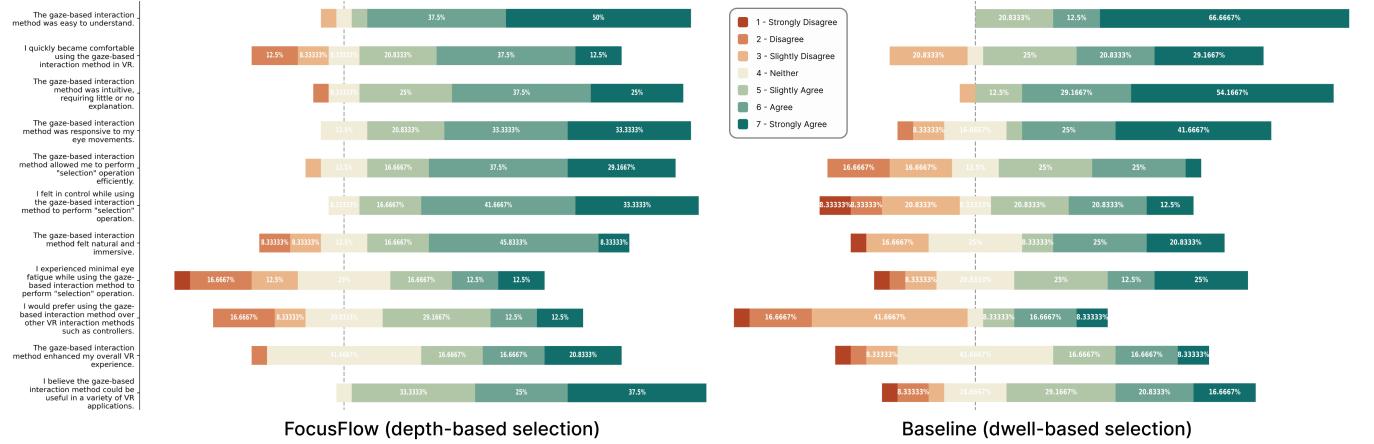


Figure 19: User Experience Likert Data.

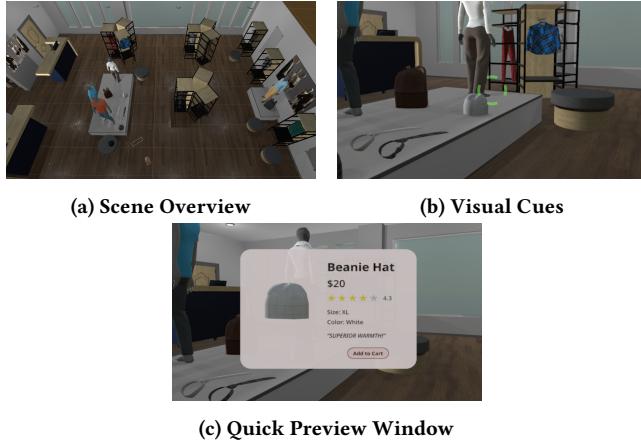


Figure 20: Case study in a shopping scenario.

However, during free exploration using the **dwell method**, users faced considerable challenges. When their gaze lingered on a product, the sudden appearance of the information panel – triggered by surpassing the observation time threshold – obstructed their view. “The most significant flaw of this method is that I can’t freely gaze at an object indefinitely. I constantly have to be cautious to not stare too long, lest the window pops up, hindering my observation. I believe the freedom to observe the world undisturbed is crucial for

user experience.” (P21) The lag between intention to inspect and the system’s response was also disruptive. “During that waiting period, even if it’s just a second, I felt anxious. It felt like wasted time since the system didn’t promptly capture my intent.” (P11) After inspecting a product, to exit the information panel, users had to move their gaze to its periphery, an action they deemed superfluous. “The dwell method required me to shift my gaze to the edge to exit. There was a continual need to oscillate between the center and periphery. In contrast, with the depth method, I merely had to naturally look past the translucent panel towards the distant environment to exit. Essentially, with the depth method, I wasn’t burdened with unnecessary actions since my goal was to resume observation, which in itself served as the exit mechanism.” (P9)

In summary, users welcomed the depth input dimension. “The depth method gave me a stronger sense of control. I could distinctly decide whether to activate a target.” (P6) “I preferred the depth method. Though time-based techniques are simpler, once I got the hang of the depth method, I noticed an improved experience.” (P8) However, they also highlighted that the depth method was more straining on the eyes after frequent use. “Compared to the dwell method, a downside of the depth approach is the fatigue it induces in my eyes due to the constant need for depth adjustments.” (P13) This feedback implies that the depth method might not be ideal for extended use in short durations, as continuous adjustments could tire the eyes.

10 LIMITATION AND FUTURE WORK

FocusFlow features a hands-free gaze-depth-based input method that is efficient and intuitive for selection tasks after a minimal learning process. We proposed a layer-based user interface design according to the visual depth estimation characteristics and learning strategies to help users learn this gaze-depth input method. Although our evaluation results demonstrate FocusFlow is superior to baseline in terms of operational efficiency and the effectiveness of the adaptive learning strategy in helping users master this gaze-depth input method, there are still some aspects that could be improved or extended in future works:

Statistical Analysis: The current paper does not include any major statistical significance analysis due to small participant population. In our future work, we plan to perform a long-term experiment with a larger user population to fully demonstrate the effectiveness and intuitiveness of the proposed approach.

Multi-modal Interaction Inputs: Visual depth is a unique input dimension with a marginal likelihood of clashing with other eye behaviors. However, it only offers limited interaction inputs such as selection or zoom-in. To improve the number of inputs, FocusFlow can be seamlessly integrated with other eye input methods, such as eyelid movements or blinks. Moreover, the visual depth modality can be harmoniously coupled with other VR interaction modes like hand and head inputs. For instance, a user might activate a virtual window through gaze-depth interaction and subsequently interact with components within the window using direct or indirect hand gestures [29]. Such compound interactions can be explored in detail in future research.

Gaze Depth as a Continuous Input: The evaluations in Section 3.3 show that high-accuracy depth estimations are limited to close distances. While this limits our accurate depth detection over long distances, we can still utilize depth data at close range for more fine-grained inputs. For example, we can expand our gaze-depth interaction from binary selection to continuous inputs by continuously tracking exact visual depth changes and mapping the depth to some operational values. This will enable various manipulations such as scrolling on a webpage [36] or rotating a 3D object [17] by looking at micro gaze-depth changes.

Virtual Window Depth Adaptability: To elevate user experience and reduce erroneous virtual window activation, the position of the interaction layer could be adapted to a target's location. For example, when the target is a few meters away from the user, the interaction layer could be set to one meter in front of the user. However, if the target distance is less than one meter, the position of the interaction layer will be automatically set to half a meter from the user. However, such adaptability necessitates visual cues to inform users of the adjusted window position, ensuring seamless interactions.

Looking Through Interaction: In this paper, we assumed that the virtual window is always placed closer to the user than the interactive object. However, FocusFlow can be seamlessly extended to applications where information is designed further from the interactive object. For example, in the surgical training application shown in Figure 21, the user can switch between observing different



(a) Looking at the object



(b) Looking through the object

Figure 21: Surgical training application.

layers inside the patient's body by actively changing their focal depth further from the upper surface of the body.

11 CONCLUSION

We present *FocusFlow*, a novel gaze-depth interaction technique that leverages the potential of visual depth as an intuitive input dimension. Our approach introduces a layer-based user interface that resonates with the concept of a “Virtual Window”, offering an efficient and hands-free selection method. By doing so, we address not only the traditional Midas Touch problem but also enhance the overall user experience in virtual reality (VR) interactions.

Considering that visual depth is a new input dimension for VR users, we place emphasis on the learnability of our design. To evaluate this aspect, we conducted both a user study and a case study to observe how novice users adapt to this new interaction method. The results demonstrate that gaze-depth interaction can be learned smoothly with adaptive visual cues that facilitate users’ perception of depth and develop muscle memory.

We hope this study will inspire more researchers to refine theoretical frameworks related to gaze-depth interaction. For instance, future work could focus on designing more effective learning processes, assisting users' depth perception during transition periods, and integrating inputs from other modalities. These efforts would provide valuable theoretical support for promoting widespread adoption of this intuitive interaction method.

ACKNOWLEDGMENTS

This research was partially funded by UIUC-Inspire program. We thank Professor Andrew Kurauchi and Professor Luciano Soares for their guidance in eye-tracking characterization and Arnav Shah for designing and implementing the surgical training application.

REFERENCES

- [1] Yasmeen Abdabou, Mariam Mostafa, Mohamed Khamis, and Amr Elmougy. 2019. Calibration-free text entry using smooth pursuit eye movements. In *Proceedings*

- of the 11th ACM Symposium on Eye Tracking Research & Applications.* 1–5.
- [2] Isayas Berhe Adhanom, Paul MacNeilage, and Eelke Folmer. 2023. Eye Tracking in virtual reality: A broad review of applications and challenges. *Virtual Reality* (2023), 1–25.
- [3] Sunggeun Ahn, Jeongmin Son, Sangyoon Lee, and Geohyuk Lee. 2020. Verge-it: Gaze interaction for a binocular head-worn display using modulated disparity vergence eye movement. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*. 1–7.
- [4] Ehsan Mohammadi Arvacheh and Hamid R Tizhoosh. 2006. Iris segmentation: Detecting pupil, limbus and eyelids. In *2006 International Conference on Image Processing*. IEEE, 2453–2456.
- [5] Yiwei Bao, Jiaxi Wang, Zhimin Wang, and Feng Lu. 2023. Exploring 3D Interaction with Gaze Guidance in Augmented Reality. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 22–32.
- [6] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. In *Proceedings of the Workshop on Communication by Gaze Interaction*. 1–9.
- [7] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbitz: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th annual ACM symposium on user interface software & technology*. 457–466.
- [8] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. 2017. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1118–1130.
- [9] Ramin Hedeshy, Chandan Kumar, Raphael Menges, and Steffen Staab. 2021. Hummer: Text entry by gaze and hum. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [10] Teresa Hirzle, Jan Gugenheimer, Florian Geiselhart, Andreas Bulling, and Enrico Rukzio. 2019. A design space for gaze interaction on head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [11] Baosheng James Hou, Joshua Newn, Ludwig Sidenmark, Anam Ahmad Khan, Per Bækgård, and Hans Gellersen. 2023. Classifying Head Movements to Separate Head-Gaze and Head Gestures as Distinct Modes of Input. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [12] Robert JK Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 11–18.
- [13] Dominik Kirst and Andreas Bulling. 2016. On the verge: Voluntary convergences for accurate and precise timing of gaze input. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 1519–1525.
- [14] Aleksandra Królak and Paweł Strumillo. 2012. Eye-blink detection system for human-computer interaction. *Universal Access in the Information Society* 11 (2012), 409–419.
- [15] Shinya Kudo, Hiroyuki Okabe, Taku Hachisu, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto. 2013. Input method using divergence eye movement. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 1335–1340.
- [16] R. John Leigh and David S. Zee. 2015. 520Vergence Eye Movements. In *The Neurology of Eye Movements*. Oxford University Press. <https://doi.org/10.1093/med/9780199969289.003.0009> arXiv:https://academic.oup.com/book/0/chapter/189897581/chapter-ag-pdf/44632480/book_25280_section_189897581.ag.pdf
- [17] Chang Liu, Jason Orlosky, and Alexander Plopski. 2020. Eye Gaze-based Object Rotation for Head-mounted Displays. In *Proceedings of the 2020 ACM Symposium on Spatial User Interaction*. 1–9.
- [18] Feiyu Lu and Doug A Bowman. 2021. Evaluating the potential of glanceable AR interfaces for authentic everyday uses. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 768–777.
- [19] Feiyu Lu, Shakiba Davari, and Doug Bowman. 2021. Exploration of techniques for rapid activation of glanceable information in head-worn augmented reality. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction*. 1–11.
- [20] Lu Lu, Pengshuai Duan, Xukun Shen, Shijin Zhang, Huiyan Feng, and Yong Flu. 2021. Gaze-Pinch Menu: Performing Multiple Interactions Concurrently in Mixed Reality. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 536–537.
- [21] Xueshi Lu, Difeng Yu, Hai-Ning Liang, and Jorge Goncalves. 2021. itext: Hands-free text entry on an imaginary keyboard for augmented reality systems. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 815–825.
- [22] Mathias N Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Gronbek, and Hans Gellersen. 2022. Gaze-hand alignment: Combining eye gaze and mid-air pointing for interacting with menus in augmented reality. *Proceedings of the ACM on Human-Computer Interaction* 6, ETRA (2022), 1–18.
- [23] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 357–360.
- [24] Diako Mardanbegi, Benedikt Mayer, Ken Pfeuffer, Shahram Jalaliniya, Hans Gellersen, and Alexander Perzl. 2019. Eyesethrough: Unifying tool selection and application in virtual environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 474–483.
- [25] Martez E Mott, Shane Williams, Jacob O Wobbrock, and Meredith Ringel Morris. 2017. Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2558–2570.
- [26] Moaaz Hudhud Mughrabi, Aunnoy K Mutasim, Wolfgang Stuerzlinger, and Anil Ufuk Batmaz. 2022. My eyes hurt: Effects of jitter in 3d gaze tracking. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 310–315.
- [27] Yun Suen Pai, Benjamin Outram, Noriyasu Vontin, and Kai Kunze. 2016. Transparent reality: Using eye gaze focus depth as interaction modality. In *Adjunct Proceedings of the 29th Annual ACM Symposium on User Interface Software and Technology*. 171–172.
- [28] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. 2012. Designing for the eye: design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*. 479–488.
- [29] Ken Pfeuffer. 2017. *Extending touch with eye gaze input*. Ph.D. Dissertation. Lancaster University. <https://doi.org/10.17635/lancaster/thesis/182>
- [30] Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdellrahman, Stefanie Meitner, Amr Saadi, and Florian Alt. 2021. ARtention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics* 95 (2021), 1–12.
- [31] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze+ pinch interaction in virtual reality. In *Proceedings of the 5th symposium on spatial user interaction*. 99–108.
- [32] Ken Pfeuffer, Lukas Mecke, Sarah Delgado Rodriguez, Mariam Hassib, Hannah Maier, and Florian Alt. 2020. Empirical evaluation of gaze-enhanced menus in virtual reality. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*. 1–11.
- [33] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE symposium on 3D user interfaces (3DUI)*. IEEE, 36–39.
- [34] Mikkel Rosholm Rebsdorf, Theo Khumsan, Jonas Valvik, Niels Christian Nilsson, and Ali Adjourlu. 2023. Blink Don't Wink: Exploring Blinks as Input for VR Games. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*. 1–8.
- [35] Naveen Sendhilnathan, Ting Zhang, Ben Lafreniere, Tovi Grossman, and Tanya R Jonker. 2022. Detecting Input Recognition Errors and User Errors using Gaze Dynamics in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–19.
- [36] Selina Sharmin, Oleg Špakov, and Kari-Jouko Räihä. 2013. Reading on-screen text with gaze-based auto-scrolling. In *Proceedings of the 2013 Conference on Eye Tracking South Africa*. 24–31.
- [37] Ludwig Sidenmark, Christopher Clarke, Joshua Newn, Mathias N Lystbæk, Ken Pfeuffer, and Hans Gellersen. 2023. Vergence Matching: Inferring Attention to Objects in 3D Environments for Gaze-Assisted Selection. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [38] Ludwig Sidenmark, Dominic Potts, Bill Bapisch, and Hans Gellersen. 2021. RadiEye: Hands-free radial interfaces for 3D interaction using gaze-activated head-crossing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11.
- [39] Nikolaos Sidorakis, George Alex Koulieris, and Katerina Mania. 2015. Binocular eye-tracking for the control of a 3D immersive multimedia user interface. In *2015 IEEE 1st workshop on everyday virtual reality (WEVR)*. IEEE, 15–18.
- [40] Vildan Tanrıverdi and Robert JK Jacob. 2000. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 265–272.
- [41] Christopher W Tyler and Maureen B Clarke. 1990. Autostereogram. In *Stereoscopic displays and applications*, Vol. 1256. SPIE, 182–197.
- [42] Boris Velichkovsky, Andreas Sprenger, and Pieter Unema. 1997. Towards gaze-mediated interaction: Collecting solutions of the "Midas touch problem". In *Human-Computer Interaction INTERACT'97: IFIP TC13 International Conference on Human-Computer Interaction, 14th–18th July 1997, Sydney, Australia*. Springer, 509–516.
- [43] Mélodie Vidal, David H Nguyen, and Kent Lyons. 2014. Looking at or through? using eye tracking to infer attention location for wearable transparent displays. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers*. 87–90.
- [44] Zhimin Wang, Yuxin Zhao, and Feng Lu. 2022. Control with vergence eye movement in augmented reality see-through vision. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 548–549.
- [45] Zhimin Wang, Yuxin Zhao, and Feng Lu. 2022. Gaze-vergence-controlled see-through vision in augmented reality. *IEEE Transactions on Visualization and*

- Computer Graphics* 28, 11 (2022), 3843–3853.
- [46] Séamas Weech, Sophie Kenny, and Michael Barnett-Cowan. 2019. Presence and cybersickness in virtual reality are negatively related: a review. *Frontiers in psychology* 10 (2019), 158.
- [47] Yushi Wei, Rongkai Shi, Difeng Yu, Yihong Wang, Yue Li, Lingyun Yu, and Hai-Ning Liang. 2023. Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
- [48] Xin Yi, Leping Qiu, Wenjing Tang, Yehan Fan, Hewu Li, and Yuanchun Shi. 2022. DEEP: 3D Gaze Pointing in Virtual Reality Leveraging Eyelid Movement. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–14.
- [49] Difeng Yu, Xueshi Lu, Rongkai Shi, Hai-Ning Liang, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2021. Gaze-supported 3d object manipulation in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [50] Zhe Zeng and Matthias Roetting. 2018. A text entry interface using smooth pursuit movements and language model. In *Proceedings of the 2018 acm symposium on eye tracking research & applications*. 1–2.
- [51] Xinyong Zhang. 2021. Evaluating the Effects of Saccade Types and Directions on Eye Pointing Tasks. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 1221–1234.

A AUTOSTEREOGRAM

Figure 22 shows an *autostereogram*, a picture can show 3D patterns at a certain visual depth.

B USER EXPERIENCE QUESTIONNAIRE

1. Usability and Learnability

- The gaze-based interaction method was easy to understand.
- I quickly became comfortable using the gaze-based interaction method in VR.

- The gaze-based interaction method was intuitive, requiring little or no explanation.

2. Responsiveness and Efficiency

- The gaze-based interaction method was responsive to my eye movements.
- The gaze-based interaction method allowed me to perform "selection" operation efficiently.
- I felt in control while using the gaze-based interaction method to perform "selection" operation.

3. Comfort and Naturalness

- The gaze-based interaction method felt natural and immersive.
- I experienced minimal eye fatigue while using the gaze-based interaction method to perform "selection" operation.

4. Preference and Overall Experience

- I would prefer using the gaze-based interaction method over other VR interaction methods such as controllers.
- The gaze-based interaction method enhanced my overall VR experience.

5. Applicability in Various Contexts

- I believe the gaze-based interaction method could be useful in a variety of VR applications.

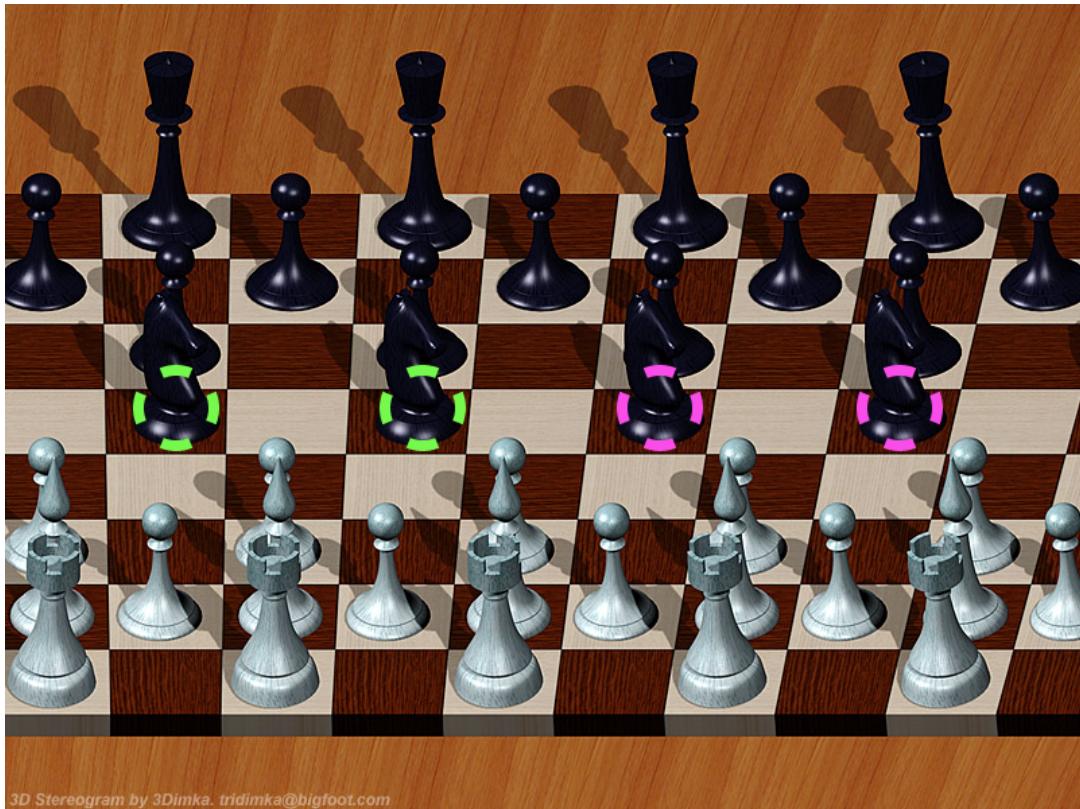


Figure 22: Autostereogram Example. When you put your gaze point slightly further and make the green circle and pink circle overlap, you can see a 3D image. Making the image as large as possible and not too close to the image will help you see the 3D image successfully.