Intuitive Interaction Methods for Smart Glasses: An Eye-tracking Based Solution

Chenyang Zhang*
zhang414@illinois.edu
University of Illinois
Urbana-Champaign
Department of Computer Science
Urbana, Illinois, USA

Rohan Nedungadi*
rohanrn3@illinois.edu
University of Illinois
Urbana-Champaign
Department of Computer Science
Urbana, Illinois, USA

Tiansu Chen*
tiansuc2@illinois.edu
University of Illinois
Urbana-Champaign
Department of Computer Science
Urbana, Illinois, USA

ABSTRACT

Extended Reality (XR) gives the user an opportunity to interact with a virtual world and is gradually becoming the next mode of interfacing with the digital world. However, compared with human natural sense in the real world, current XR applications and their interaction methods are counter-intuitive with limited ways of input. Current interaction methods, such as head-tracking and controller-based ones, can cause fatigue through body movements with significantly high response time during interaction with the XR objects, resulting in an unpleasant user experience when using the XR device. In this paper, we propose a gaze-based interaction solution in the XR space, aiming at providing a more natural and straightforward interactive experience for XR users. We design a systematic pipeline with three main modules to map the user's gaze to the XR system response. In addition, we introduce a validation step in order to reduce potentially dissatisfying incorrect responses. With the experiments, we demonstrate that our intuitive gaze interaction method is able to improve the efficiency and accuracy of the interaction process. Moreover, the result shows the significance of the validation step and the decrease in human motion compared with the head-tracking interaction.

KEYWORDS

interaction design, augmented reality, eye-tracking, embedded system

ACM Reference Format:
Chenyang Zhang, Rohan Nedungadi, and Tiansu Chen. 2022. Intuitive Interaction Methods for Smart Glasses: An Eye-tracking Based Solution. In *Proceedings of Smart-X '22: Fake ACM Symposium on Smart cities, homes, phones, and beyond (Smart-X '22).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the past twenty years, mobile devices have profoundly changed the way we receive and send messages. The reason behind this is the giant leap in interaction design [6, 46, 47] based on the multi-touch technologies [35], enabling us to interact with these portable devices easier and more efficiently. Compared to traditional mobile devices, Augmented Reality (AR) and Virtual Reality (VR) glasses take the efficiency of information delivery to a whole new level, i.e. by providing a larger array of information in a more immersive way, almost as if bringing it within our reach. This is achieved by showing all content in the view of a 3D world, instead of being limited by fixed-sized and device-constrained screens.

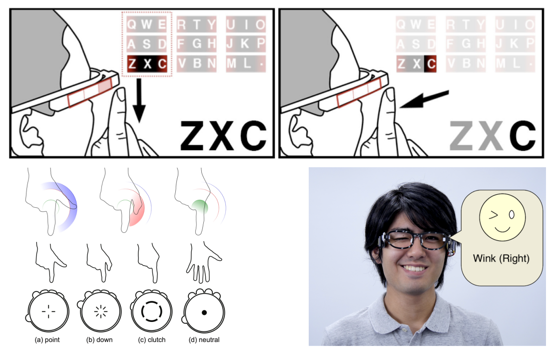


Figure 1: Existing interaction methods for smart glasses. SwipeZone [15] (Top, touch-input), Gunslinger [41] (Bottom-left, gesture-input) and Facial Expression Recognition [44] (Bottom-right, facial-input). These new interaction methods try to balance the input efficiency and portability, however, they all suffer from usability problems.

Accordingly, customized interaction methods that cater to this new approach are expected to see a rise. Research in this area has picked up steady growth and will continue to advance in the near future, in order to unlock the efficiency of the new information platform, else, users may suffer from information overload or underload problems. As Figure 1 shows, many interaction methods with touch [15] or touchless [41, 44] input have been proposed [34] for smart glasses. However, these interaction methods are either inefficient or difficult to learn, making smart glasses an immature device for this day and age, thereby to some extent preventing these products from being accepted by the general public. According to cognitive science research [7], humans are best adapted to use eye movements to control attention shifts and thus target different objects using the eye gaze. The reason behind this is simply that the human brain is able to easily apply focus and attention to visual information that is directly transferred from the eye, hence motivating us to explore the use of eye-tracking technology to build attention-based interactions. On the other hand, with advances in micro-tech the eye-tracking hardware [27] has become sufficiently small and light enough to be integrated into the Head Mounted Displays (HMDs), allowing us to track the eye movements on smart glasses and then

*Both authors contributed equally to this research.

Smart-X '22, Jan 18–May 05, 2022, Urbana, IL
2022. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

performing interaction design. Applying eye-tracking in AR interactions is feasible and promising.

This research aims to explore an efficient and intuitive gaze interaction method for Extended Reality (XR) HMDs, improving the accessibility and controllability of information elements while reducing the inefficient "physical-motion" (i.e. head rotation or arm movement) cost of human input on HMDs. Throughout the process, we will maintain an observational study with the target user group, finding their needs and expectations with XR interaction, and collecting feedback to guide the design iterations. We believe that gaze interaction approaches are expected to lead to better efficiency than the now widely used head-tracking methods. Thorough evaluations are conducted in this paper with several metrics such as accuracy, precision, robustness, and latency to provide a fair comparison between the two solutions. While the proposed solution is mainly developed and tested on a VR headset, the approach is extensible to AR head sets as well.

Although some researches and product solutions have approached the Human-Computer Interaction (HCI) problem with gaze interaction, they have had their fair share of limitations and issues. Consumer Electronic Products like *VIVE Pro Eye* (VR headset) and *Magic Leap* (AR headset) integrate eye-trackers and do provide APIs for the developers and researchers, but the applications are limited to data collection and naive use cases [63]. There have been promising interaction design research that proposed some eye-tracking based mechanisms [43, 51, 53], but their interaction method is not well-designed, leading to some rough prototypes with a lack of detailed evaluations.

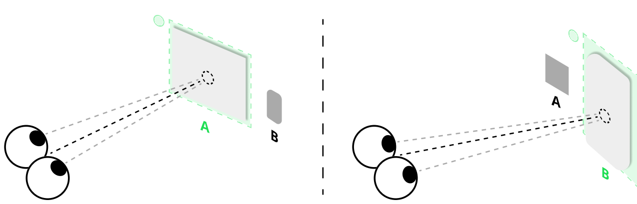


Figure 2: Activating an object through our gaze interaction. The user only need to focus on the target object target object, then the object will be activated automatically, just like the "hover" effect on the computer screen.

In this paper, we propose a gaze interaction system in the XR platform. Basically, the system contains three main steps in each time cycle, including capturing the eye movement, analyzing the interaction and executing in the virtual world. First, we build a virtual world with several objects in Unity as the simulation of XR environment. In order to track user's eye movement as the system input, we adopt *VIVE Pro Eye* headset as our hardware, which is integrated with the eye-tracking functionality recognizing user's current gaze information when interacting with the virtual world. With the input gaze data above, further analysis can be done to determine the expected system response in the interaction according to predefined interaction rules. More specifically, since the interaction happens in real-time, the system should be aware of the response of each object at each time step in the virtual world to meet user's expectation in the interface, which indicates that both object selection and response matching are required during the analysis. Finally, the system updates the virtual world through executing the response for each object, so that the user is able to interact with the virtual world using their eye movement.

In our proposed gaze interaction system, we are faced with several research challenges regarding the effectiveness and the efficiency of the interaction process. Since eye movement is subtle, while the interactive objects in the XR world may be closely packed, it requires a precise method to locate the gaze and map it into the virtual space so that users can make effective interactions with the XR world. Moreover, since the user relies on eye movement for perceiving and interacting with the virtual world, the system needs to minimize the number of unexpected interactions by distinguishing the interactive gaze behaviors, so that the user will have better interaction experience.

Several experiments were conducted to compare the performance difference between gaze interaction and the no-gaze (head) interaction method. First, to evaluate the efficiency of gaze interaction, we gave users a well-designed task and recorded the speed of operations in the process. Then, the accuracy of the gaze pointer was evaluated in another task, which can tell the distribution of hitting points on the target object. Moreover, we evaluated the effectiveness of our validation module by analyzing false triggers and the error rate in the operations. In addition, to study the user behavior using the new interaction method, the head rotation was visualized in heatmaps that reveal different usage processes between the gaze and head interaction methods. Finally, a qualitative study was done to test whether gaze interaction is intuitive and easy to learn for most people. Experimental results show that our method achieves a low error rate while being efficient, and the validation module is necessary for usability. Also, qualitative research shows the gaze interaction is intuitive and easy to learn.

In summary, the contributions of this work are:

- A well-designed gaze interaction system, enabling users to interact with virtual objects in an intuitive, efficient and accurate way.
- An effective validation step that greatly reduces the incorrect system responses during gaze interaction process.
- Detailedly designed use cases and rigorous evaluation with both quantitative and qualitative results compared with the baseline method.

2 RELATED WORK

2.1 Interaction Methods for XR HMD

Although extended reality and smart glasses just became a development hit in the last decade, they have been evolving for a long time since the last century [13, 69]. Lee and Hui [34] review the interaction methods have merged so far and provide a taxonomy of these techniques, which classifies the interactions into handheld, touch, and touchless based on input methods.

Handheld controller input is the most commonly used interaction method for today's commercial XR products. These handheld devices can provide accurate and low-latency input data such as the hand position and button click to the virtual world, however, these handheld controllers are inconvenient to carry with, so their applications are limited in the entertainment field [19] and some professional use cases [24, 61].

Touch is an input approach inherited from today's mainstream interaction method for mobile devices. As XR HMDs do not have a touch screen, the touch input method evolved into on-body touch and on-device touch [34]. The on-body touch facilitates the human body, such as palm [68], forearm [48], finger [71], face [16], and ear [39], as a medium of interaction. These interaction media are easy to access and can receive complex instructions through subdividing the operation area [22], but it is difficult for users to remember the mapping between the touch area and system response. On the other hand, the on-device touch is an input method using the touchpad on HMD [12] or extra wearable devices, like ring [17, 66] or wristband [36, 60]. These input devices are more portable than traditional controllers, but the design space for them is limited due to their restricted input dimension or relatively low expressiveness.

Touchless input is a relatively new method benefits from the development of Computer Vision technologies. The hands gesture-based interactions [2] have become a widely used method, which detects the hand gestures through HMD built-in cameras, and translates the image patterns to interaction input. This interaction method is intuitive to use since we can manipulate virtual objects with familiar gestures in reality such as click, swipe, grasp, zoom, and drag [3, 4, 29, 72]. But moving hands in the air for a long time tends to make users feel tired, so other hands-free methods based on head movement [1], gaze movement [56], or voice recognition [57] also raised people's interest. These hand-free methods are intuitive and easy to use, but how to ensure the accuracy and efficiency of interaction is a non-trivial problem.

In summary, efficiency, expressiveness, intuitiveness and portability are the main concerns for the XR HMD interaction methods. Also, we need to consider both the generality of the method and its applicability to different application scenarios when designing interaction methods.

2.2 Gaze Interactions

The development of eye-tracking technology enables gaze-based interactions in multiple platforms. From the last century, the gaze-input has been studied as an HCI problem [23]. With the evolution of personal computing devices, gaze interaction research has been conducted on computers [28], mobile phones [54], tablets [20], and head-mounted displays [5]. However, only few of these researches are transferred to widely used products for non-research use.

The reasons for this phenomenon include a number of interaction design problems [49]. For example, the Midas Touch effect [25] is a classic user experience (UX) problem about how to distinguish between an intentional fixation (should trigger an element's action) and an accidental one (should have no effect). In gaze interaction, it is especially important to judge the intention of eye movements. In addition, many researches only focus on the possibility of using eye-tracking alone to build a new interaction method [62], resulting in the loss of design space, making this technology mostly used for accessibility design. Also, the screen size can be a constraint for the value of gaze interaction, since the need for gaze input is low in small-screen devices that can be easily operated by hand.

Extended Reality creates an all-new design space for interaction design that shifts the dimension of the user interface from 2D (all elements on a plane) to 3D (elements in a virtual world) [56]. This provides a good opportunity for gaze interaction in XR HMD. First, the gaze, or attention, is the most intuitive and natural way for humans to see and target an object [33], which guarantees that most people can easily learn this interaction method. Second, the eye-movement is the fastest way to switch viewing points and observe in a wide area, which means the gaze interaction can be more efficient than the method based on head or hand movement. In addition, the eye-tracking module can be built in the HMD with high accuracy [26, 64], ensuring the portability of hardware and accuracy of gaze recognition.

Some exploratory work on XR gaze interaction has been conducted in recent years. Hirzle et al. [18] discussed the design space for gaze interaction on HMD, mapping the techniques to the application scenarios. Piumsomboon et al. [55], Piening et al. [53] and Choi et al. [9] proposed gaze-only XR interaction methods, which basically make use of the viewpoint position and dwell time to do operations like click, hold, and move. Beyond this, some multi-modal gaze interactions are created to better facilitate complex tasks. Pfeuffer et al. [51, 52] and Lystbæk et al. [43] combined the eye and gesture as the input, improve the efficiency and expand the functionality of gaze XR interactions. Some researchers also fine-tuned the interaction method from the algorithm side to enhance the user experience. Kytö et al. [32] demonstrated a proof-of-concept interaction method for online correction of calibration drift with modalities of head motion and eye gaze. These researches revealed the feasibility of gaze interactions, but most of them lack rigorous quantitative analysis and well-defined use cases. It is still hard for designers to develop a design language based on these gaze interactions.

In our work, we establish a comprehensive gaze interaction system and conduct rigorous analysis with well-defined metrics. To fill the gap between academics and applications, we further design several use cases to show the effectiveness of our interaction method and guide the future design practice.

2.3 Applications of XR Interactions

Immersive applications have become a new hot spot with the increasing popularity of XR devices. In general, the applications can be categorized into professional use and public use.

The most significant requirement of professional applications is providing abundant and accurate data that can help experts make rational analyses. Lin and Yang et al. [38] investigated immersive analytics in sports. For example, the system can give real-time feedback in basketball training [37]. Some researches studied the XR application in surgery [30, 50], to assist the surgeon in training, planning, and operating process. Some professional applications in 3D modeling [11, 42] and construction [65] are also promising. To collect and observe high-quality data in the above scenarios, multi-modality interaction methods are needed to catch more subtle input actions. Therefore, the basic XR interaction methods should have good compatibility to work with supplementary devices in professional use cases.

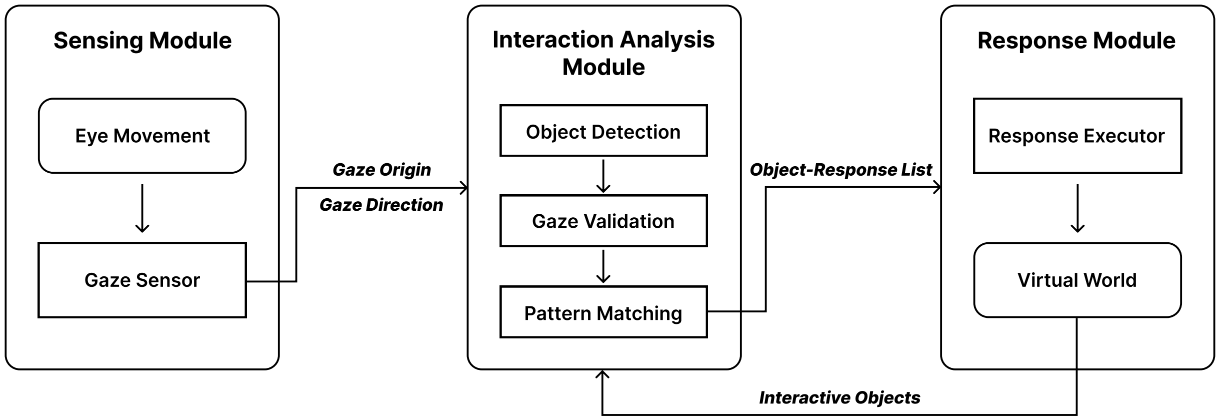


Figure 3: System overview. Sensing Module captures user's eye movement and converts it into vectors. Interaction Analysis Module determines the interacted objects and analyzes their expected responses. Response Module executes the responses and updates the virtual world.

In contrast to the rigor of experts, applications for the general public need to put more emphasis on usability and acceptability. XR headsets have been used in the astronomy [45], geography [58], and engineering [21] educations to give students more vivid observations. In the field of media art [14] and entertainment [8], XR offers new creating and viewing approaches. In terms of journalism and social media, the XR technologies are used to help the readers gain first-person experience [10, 31]. Researches from immersive data visualization [40, 70] also provide an opportunity for improving public data literacy and science popularity. The applications for different people in many fields illustrate the necessity to develop a consistent design language in XR. Hence, the interaction methods need to be general enough for maximum public acceptance.

In conclusion, XR applications face both professional and popular users, and can be used in rich scenarios to show a large amount of interactive information. To work in more scenarios, the gaze interaction needs to be both compatible with multi-modal input, and general enough to be accepted by the public.

3 SYSTEM OVERVIEW

This section explains how the eye tracking application is integrated into the virtual world and how to define interaction between our gaze and objects in the virtual world. Figure 3 presents an overview of the gaze interaction system for a single time step, which consists of three main modules, converting the physical eye movement into the system response in the XR space.

In the Sensing Module, the gaze sensor captures the eye movement and converts it into two vectors, gaze origin and gaze direction. The former vector describes the starting point of user's gaze in the XR space, while the latter is a unit vector describing the direction of user's gaze from the origin. These two vectors are then transmitted to the next module for further analysis.

The next module is called the Interaction Analysis Module, which analyzes the expected response for each object in the virtual world. The analysis process is divided into three steps, corresponding to the submodules in the figure. In the Object Detection, with the access to the interactive objects in the virtual world, the system first detects the target object hit by the current gaze. Since only some meaningful gaze behaviors should trigger the interaction process, we include the Gaze Validation submodule to distinguish the valid gazes from all eye movements. The last submodule, defined as Pattern Matching, matches each object with its response action according to its current and expected status. The details of these submodules will be discussed in the section 4. In the end, this module outputs a list containing the expected response for each object in the virtual world.

Finally, with the *Object-Response List*, the Response Module executes the response of each object and therefore updates the virtual world. As a result, the user is able to obtain real-time response with the gaze interaction.

4 PROPOSED APPROACH

In this section, we discuss our proposed approach to design the gaze interaction system, and explain our motivation for each submodule. This section is divided into three parts, corresponding to the system architecture in Figure 3.

4.1 Sensing Module

In order to capture the eye movement, the camera and gyroscope sensors are required to track the shifting gaze and the changing view in the XR space. However, with the gazes from two eyes respectively, we need to find a way to combine them and define the gaze attention, so we adopt the open-source eye-tracking software called HTC Super Runtime animation pal (SRanipal) SDK [67]. Figure 4 gives an overview of how the HTC SRanipal SDK computes the combined gaze direction by tracking each individual eye. First, the eye tracker essentially tracks each eye's deviation from the origin individually with respect to its own coordinate system where the origin is the center of the eye lens. It then determines each eye's gaze direction and finds the intersection of these two directions to determine the focus point. Finally, the vector from the system origin, which is the center of the eyes, to the focus point, determines the final gaze direction in the XR space.

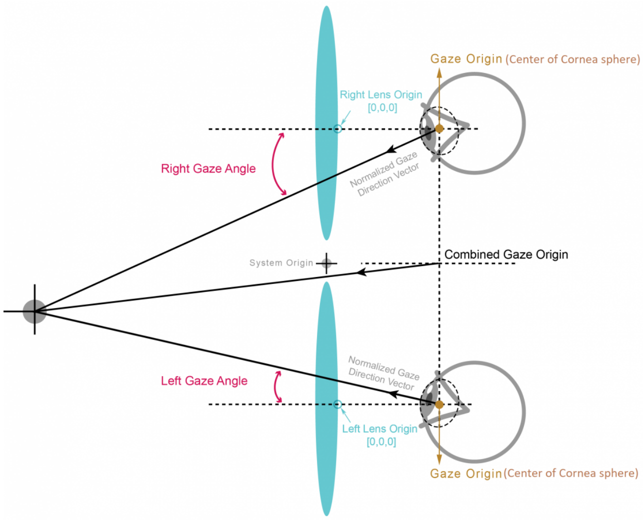


Figure 4: Combined gaze analysis in SRanipal [67]. The combined gaze is determined by the intersection of individual gaze from each eye.

4.2 Interaction Analysis Module

With gaze origin and gaze direction information, the system needs to understand user's expected interaction through the gaze behavior. So this module is designed for interaction analysis, which includes three submodules for recognizing interacted objects (Object Detection), checking the validity of gaze behavior (Gaze Validation) and matching them with expected interaction patterns (Pattern Matching). In the end, the result list containing the response for each object is sent to Response Module for execution in the XR space.

4.2.1 Object Detection. As the first step in the interaction analysis, we need to first determine the current interacted object in the XR space. In our approach, we apply a ray casting method to locate the target object. Since the system records the gaze origin and gaze direction information from the Sensing Module, we can extend the gaze direction as a ray and check if the ray hits any object. If it does, the hit object is recognized as the target object in the current time step.

4.2.2 Gaze Validation. Before the analysis moves forward, we include another step to improve the interaction efficiency. Since eye movement happens all the time for human, one of the most significant differences between eye-tracking based interaction method and other methods is that the former one is more ambiguous. When the users are interacting using body gestures or controller devices, the system should respond to each of the interactive behavior since users are totally subjective to start an interaction. However, in an application with gaze interaction system, the system should not make any response when the user is just moving the eyes to perceive the world (Figure 5).

To solve this problem, we design a validation step before starting matching the interaction patterns. The main purpose of this step is to enable the gaze interactions only when the users intend to interact with their eyes in order to avoid potential incorrect system responses which may dissatisfy the user experience. According to our experiments in 6.3, the validation step can improve the effectiveness of the gaze interaction.

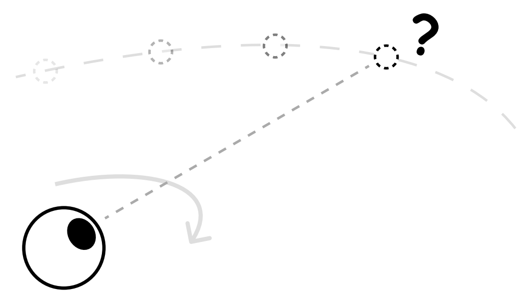


Figure 5: Figure needs to be more specific. Motivation of gaze validation. Human eye keeps moving all the time, so we need to determine whether the system should give any response or not when detecting an eye movement.

4.2.3 Pattern Matching. In order to expand the diversity of eye-tracking based interactions, the system should make different responses to different gaze behaviors, and the type of target interactive objects should also decide the way of objects to respond. So the Pattern Matching submodule is designed to determine the pattern of the interaction. Basically, it takes both current gaze information and objects into consideration, and maps them into a predefined interactive pattern for subsequent response processes, such as zooming, shifting, changing color and so on. Moreover, this submodule also takes charge of assigning some restoring responses to the objects that interacted with in the previous time steps. The variety of interactive patterns makes it possible for users to perform appropriate interactions under different circumstances, and finally enrich the user experience during the interaction process.

4.3 Response Module

To perform the interaction in the XR space, we create a virtual world as the test bed for designing and evaluating our approaches. which includes a background scene and interactive objects. All the interactive objects are capable of making responses to user's gaze interactions. After analyzing the interactions in the previous module, the Response Executor submodule executes the response by modifying the objects in the virtual world, so that the user is able to see the response of their interactive behaviors in real time.

5 IMPLEMENTATION

In this section, we discuss about both hardware and software components in the gaze interaction system, and specify the implementation details of some important modules.

5.1 Hardware

For the Sensing Module, we adopt the *HTC Vive Pro Eye* headset with a 110° field of view, 1440×1600 pixels resolution per eye and 90 Hz refresh rate. It is also integrated with an eye-tracking module capturing the gaze data with 120 Hz output frequency.

5.2 Software

For the software implementation part, we adopt the Unity 3D game engine to build the gaze interaction system. In the following sections, we explain our detailed implementation of the submodules of the key component in the system architecture, the Interaction Analysis Module.

5.2.1 Ray Casting Detection Method. To realize the ray casting method for detecting current interacted object, we adopt a 3D Physics model in Unity called RayCast [59]. The RayCast object is part of the Physics module in Unity, mimicking the laws of a light ray, which is frequently used in Unity to detect other objects in the world or perform asynchronous object communication (such as collision). Once the Interaction Analysis Module receives the gaze origin and direction information from the Sensing Module as well as the interactive objects from the Response Module, it can create a RayCast object based on the gaze information with infinite length. Then the RayCast object casts a ray which can collide with the interactive objects in the virtual world. As a result, the target object can be determined by the collision result of the RayCast object.

5.2.2 History Gaze Based Validation. As mentioned in section 4.2.2, a validation step is required to avoid potential incorrect system responses. In our implementation, we define two concepts for users' gaze, which are *stable* and *focused*, and the definition is shown in Figure 6. Besides the current gaze, we introduce the 'average history gaze', the origin and direction of which is defined as the average gaze origin and gaze direction in a certain number of previous time steps. Then, the current gaze is *stable* if and only if the deviation of the current gaze and the average history gaze is within a specified bound, and the current gaze is *focused* if and only if it hits the same object with the one hit by the average history gaze. The current gaze is *valid* if and only if it is *stable* and *focused*, and only a valid gaze can trigger a new interaction on objects in the XR space.

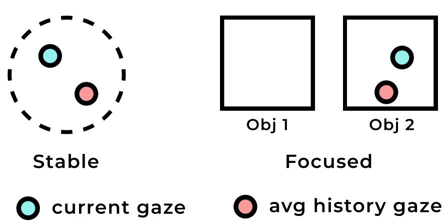


Figure 6: Definition of *stable* and *focused*. Stable: The deviation of two gazes is within a specified bound. Focused: Two gazes hit the same object.

The reason we divide the validation into two aspects is that we want to minimize the number of false triggers and maintain the sensitivity of the gaze interaction at the same time. With the concept of *stable*, it is guaranteed that large-scale eye movements won't trigger the interaction process on hit objects. On the other hand, the concept of *focused* helps improve the precision of the gaze interaction by tightening the conditions for triggering the interaction. In section 6.3, we evaluate the system performances with different validation parameters to find the best validation setting in Figure 11.

5.2.3 Interaction Pattern Design. For the interaction patterns in the proposed gaze interaction system, we introduce three states for each object, including *Default*, *Active* and *Hover*. While the first two are straightforward to be defined as the default deactivated state and the activated state respectively, the *Hover* state is included to smooth the interaction process. Since the gaze validation step requires a small difference between the current gaze and the average history gaze, the user cannot receive any feedback until the current gaze is *stable* and *focused* enough to trigger an interaction. In order to improve real-time user experience, the *Hover* state is designed to provide an immediate response when the user's current gaze hits the objects in the XR space. Inspired by common user interfaces on the computer, the *Hover* state is defined as highlighting the target object, making the user aware of the object that is about to be activated.

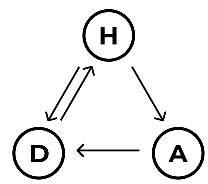


Figure 7: Transition map between object states. D for *Default*, A for *Active*, H for *Hover*.

The transition map between different states is shown in figure 7, and the transition conditions are defined as follows:

- **D → H**: The object is hit by the current gaze.
- **H → D**: The object is no longer hit by the current gaze.
- **H → A**: The current gaze is *valid* (e.g. *stable* and *focused*).
- **A → D**: The current gaze is not *focused*.

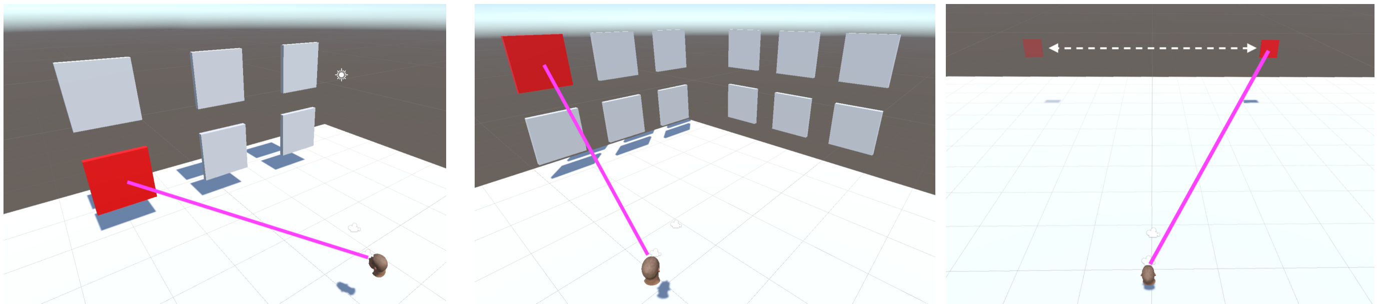
Since the purpose of the *Hover* state is to provide an indicator of the current gaze attention for the user, the conditions of transitions between *Default* and *Hover* are based on the hit result of the current gaze. As mentioned in section 5.2.2, the interaction can only be triggered by a *valid* current gaze. Finally, to restore an object from *Active* back to *Default*, the condition only cares about whether the current gaze is *focused*, because the gaze may not be *stable* when browsing an activated object. In contrast, if the current gaze is not *focused*, it indicates that the user has removed the attention from the *Active* object.

6 EVALUATION

In this section, we first describe the details of our experiment setting, then show the quantitative evaluation result to answer several usability problems and discuss the qualitative analysis result to prove the gaze interaction approach is intuitive to users.

6.1 Experimental Setup

To evaluate the accuracy and efficiency of the gaze interaction method, we designed three tasks that users can do with gaze interaction and head-tracking interaction.



(a) **T1. Selecting targets in a plane.** (b) **T2. Selecting targets in 3D space.** (c) **T3. Following the moving target.**

Figure 8: Three evaluation tasks in Unity.

6.1.1 Baseline. To some extent, gaze interaction is an enhancement of head-tracking interaction — like adding a 2-DoF pointer to the head. Therefore, the head-tracking interaction which only detects the head movement is an appropriate baseline to be compared with the gaze interaction method. Specifically, head-tracking emits a ray that always points to the center of the user's field of view, and the user will control the ray to interact with objects.

6.1.2 Conditions. To compare the difference in efficiency with the head-tracking method and emphasize the effectiveness of the gaze validation, we set up four experimental conditions as 1) gaze-tracking with validation, 2) gaze-tracking without validation, 3) head-tracking with validation, 4) head-tracking without validation.

6.1.3 Tasks. Considering the different application scenarios, we designed the following three tasks:

- **T1. Selecting targets in a plane.** In this task (Figure 8a), six squares are evenly spaced on the plane in front of the user, and one random square will become the target (the red one). The user is asked to select the target object using the gaze/head ray. After the red target is selected by the user, a new square will be the target object. This process will be done in an iterative fashion. the user is asked to
- **T2. Selecting targets in 3D space.** In this task (Figure 8b), we add one more plane of squares to the scene based on T1, aiming to expand the operation space to observe users' behavior in scenarios closer to virtual reality. The user is asked to interact with these cubes the same as T1.
- **T3. Following the moving target.** In this task (Figure 8c), a moving target placed in front of the user is moving repeatedly left and right. Users are asked to follow the center of the moving object using their gaze/head ray. This task can be used to measure the accuracy of different interaction methods when tracking a moving object.

6.1.4 Participants. As the course time is limited, we collect the quantitative data by testing the system ourselves for the completeness of the project, which is not convincing and should be optimized. In the future, we can recruit participants of different ages to conduct a more rigorous and extensive user study.

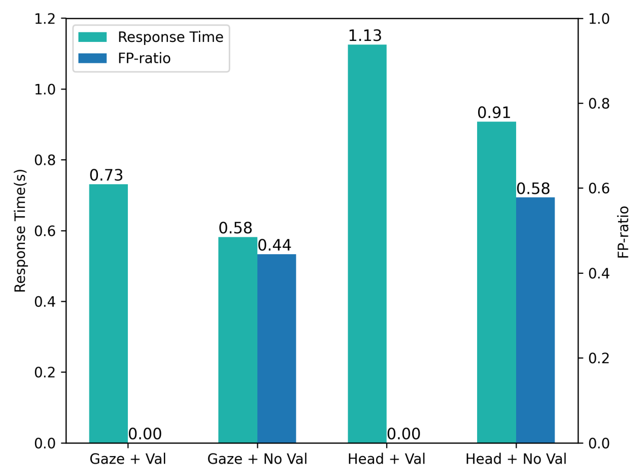


Figure 9: The system performance measured by the response time and FP-ratio in different experiment settings. The experiments are conducted on different combinations of gaze-tracking vs. head-tracking and with validation vs. without validation.

6.2 Is using gaze beneficial?

6.2.1 Efficiency. The efficiency of interaction can be reflected by the speed of operations. In our task (**T2**), the efficiency is measured by the average time to select the target object. The green bars in Figure 9 show that time under different settings (gaze/head, w./w.o. validation), where we can tell the response time of gaze interaction is 36.3% lower than the head-tracking method when without validation, and 35.4% lower when with validation. It is obvious that gaze interaction is faster and therefore more efficient than the head-tracking approach.

6.2.2 Accuracy. We measure the accuracy of gaze and head interaction in a moving scenario (**T3**). Figure 10 shows the distribution of hit points using two different methods, where each red point is a hit event to the target object. The hits with gaze interaction (left) are more concentrated toward the middle of the target, and the error rates (variance: 0.0363, bias: 0.2520) are lower than the hits with head-tracking (variance: 0.0668, bias: 0.3731), proving the gaze interaction is more accurate for the moving scenarios.

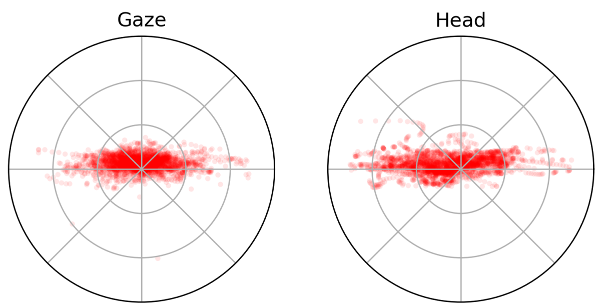


Figure 10: Visual representation of gaze-tracking vs. head-tracking method in T3. The scatter plot represents the activations (hits) triggered by the tracker with gaze vs. with head.

6.3 Validation or Not: Response Time vs. Error Rate

In section 5.2.2, the validation step is based on the difference between the current gaze and the average history gaze. Intuitively,a longer history will lead to a lower FP-ratio and longer response time, since it tightens the standard for the validation. To find the optimal validation setting in the trade-off, we explore the system performance with different lengths of gaze history, i.e., the different number of previous frames taken as history. Figure 11 shows that, with an increasing number of history frames, the FP-ratio decreases and the response time increases as expected. However, we find that when increasing the history frame number from 25 to 30, the FP-ratio sharply goes down to 0, while the response time only rises slightly. Hence, we take 30 as the optimal history frame number for the validation step.

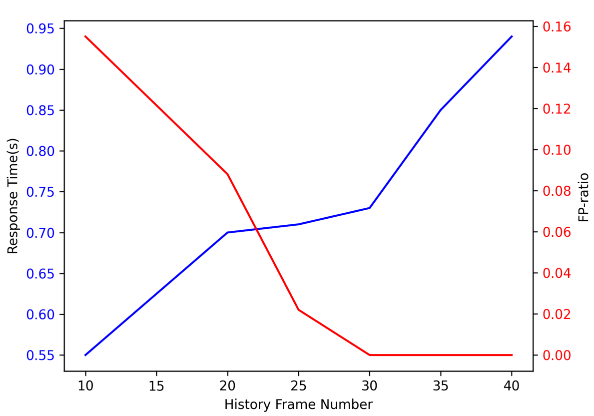
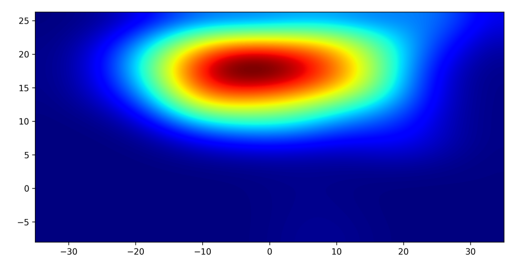


Figure 11: Calibrating the validation module. The trade-off between response time and FP-ratio with different history frame numbers.

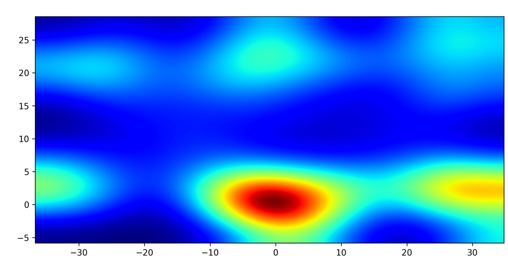
We evaluate the effect of validation under the optimal history frame number setting. Although the response time (Figure 9 green bars) is longer when applying the validation, the false triggers (Figure 9 blue bars) are significantly reduced. To be specific, for the gaze method, the response time increased by 20.5%, but the error rate decreased from 44% to 0, indicating the validation achieves a good result with a small cost.

6.4 How did participants move and interact?

How will the new interaction method affect users' behavior? We tracked the user's head movement in T1 and visualized the distribution of head pointing direction on the object plane as a heat map. Comparing the results of gaze interaction and head-tracking, it can be concluded that the gaze method allows the user's head to remain in a relatively stable range with smoother motion, while head-tracking causes a large movement of the user's head, which is demonstrated by the multiple peaks in the heatmap ().



(a) gaze-tracking interaction



(b) head-tracking interaction

Figure 12: Head motion in Gaze Tracking vs Head Tracking. The heat map (red - dense, blue - sparse) displays the density of head vectors (representing the motion of the head) in T1.

6.5 Is gaze interaction intuitive enough?

One of the primary goals of our design is to be intuitive for the general public. To evaluate this, we conducted some qualitative data collection through a roadshow with 17 students. During their use, we only told them to try to use their eyes to operate, other than that we didn't tell them any other information. All of them gave positive comments to our interaction design, "*The overall idea is easy to follow*", "*The design is very simple and yet extremely effective*", "*It works like magic*". In addition, they acknowledged the necessity of gaze validation, "*the demo shows off the benefits of validation very well*", "*It can be dizzying without validation*". The feedback received showed that gaze interaction can be easily learned and the design meets functional completeness. However, more experiments should be done in the future with participants of different ages since students are more familiar with using electronic devices.

7 CONCLUSION

In this paper, we propose the use of eye-tracking as a new and revolutionary method for intuitive interactions in the XR space. We introduce a significantly improved interaction method for XR by augmenting eye-tracking to existing approaches with a systematic pipeline that leads to shorter response time, less unwanted system responses and higher precision in object interaction than previous techniques such as head-tracking in the XR domain. Furthermore, our approach of introducing the validation layer in the interaction analysis process greatly reduces false positives at a minor expense of increased response time, making eye-tracking a much more practical solution. On the other hand, the pattern matching layer provides the interaction system with higher extensibility for multiple patterns. Through the evaluation of multiple tasks, we demonstrate that using gaze interaction along with our designed system has a positive impact on the user experience compared with the existing methods. We believe that our experiments show the significance of introducing the eye-tracking technique and the possibility of the gaze interaction becoming the next SOTA interaction paradigm for XR, hence opening many avenues for further research and novel solutions.

REFERENCES

[1] Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. 2016. Interaction techniques using head gaze for virtual reality. In *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 110–114.

[2] Daniel Bachmann, Frank Weichert, and Gerhard Rinkenauer. 2018. Review of three-dimensional human-computer interaction with focus on the leap motion controller. *Sensors* 18, 7 (2018), 2194.

[3] Abdelkader Bellarbi, Samir Benbelkacem, Nadia Zenati-Henda, and Mahmoud Belhocine. 2011. Hand gesture interaction using color-based method for tabletop interfaces. In *2011 IEEE 7th International Symposium on Intelligent Signal Processing*. IEEE, 1–6.

[4] Mark Billinghurst, Tham Piumsomboon, and Huidong Bai. 2014. Hands in space: Gesture interaction with augmented-reality interfaces. *IEEE computer graphics and applications* 34, 1 (2014), 77–80.

[5] Andreas Bulling and Hans Gellersen. 2010. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing* 9, 4 (2010), 8–12.

[6] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-" touch" interaction around small devices. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 201–204.

[7] Kang-chen Chen and Hye Jung Choi. 2008. Visual attention and eye movements. *Donald Bren School of Information and Computer Sciences (ICS). Estados Unidos: Universidad de California en Irvine. munication* 5, 1 (2008), 65–93.

[8] Adrian David Cheok, Michael Haller, Owen Noel Newton Fernando, and Janaka Prasad Wijesena. 2009. Mixed reality entertainment and art. *International Journal of Virtual Reality* 8, 2 (2009), 83–90.

[9] Myungguen Choi, Daisuke Sakamoto, and Tetsuo Ono. 2022. Kuiper Belt: Utilizing the "Out-of-Natural Angle" Region in the Eye-Gaze Interaction for Virtual Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 357, 17 pages. https://doi.org/10.1145/3491102.3517725

[10] Nonny De la Peña, Peggy Weil, Joan Llobera, Bernhard Spanlang, Doron Friedman, Maria V Sanchez-Vives, and Mel Slater. 2010. Immersive journalism: Immersive virtual reality for the first-person experience of news. *Presence* 19, 4 (2010), 291–301.

[11] Francesco Fassi, Alessandro Mandelli, Simone Teruggi, Fabrizio Rechichi, Fausta Fiorillo, and Cristiana Achille. 2016. VR for cultural heritage. In *International conference on augmented reality, virtual reality and computer graphics*. Springer, 139–157.

[12] DB Forinash. 2015. Google glass. *calico journal* 32, 3 (2015), 609–617.

[13] Michael A. Gigante. 1993. 1 - Virtual Reality: Definitions, History and Applications. In *Virtual Reality Systems*, R.A. Earnshaw, M.A. Gigante, and H. Jones (Eds.). Academic Press, Boston, 3–14. https://doi.org/10.1016/B978-0-12-227748-1.50009-3

[14] Yingjun Gong. 2021. Application of virtual reality teaching method and artificial intelligence technology in digital media art creation. *Ecological Informatics* 63 (2021), 101304.

[15] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Copenhagen, Denmark) *(MobileHCI '15)*. Association for Computing Machinery, New York, NY, USA, 144–152. https://doi.org/10.1145/2785830.2785867

[16] Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. Facetouch: Enabling touch interaction in display fixed uis for mobile virtual reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 49–60.

[17] Teng Han, Fraser Anderson, Pourang Irani, and Tovi Grossman. 2018. Hydroring: Supporting mixed reality haptics using liquid flow. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 913–925.

[18] 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.

[19] Philipp Hock, Sebastian Benedikter, Jan Gugenheimer, and Enrico Rukzio. 2017. Carvr: Enabling in-car virtual reality entertainment. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4034–4044.

[20] Corey Holland and Oleg Komogortsev. 2012. Eye tracking on unmodified common tablets: challenges and solutions. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. 277–280.

[21] Ildiko Horvath. 2016. Innovative engineering education in the cooperative VR environment. In *2016 7th IEEE international conference on cognitive infocommunications (CogInfoCom)*. IEEE, 000359–000364.

[22] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. Digitspace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1526–1537.

[23] Thomas E Hutchinson, K Preston White, Worthy N Martin, Kelly C Reichert, and Lisa A Frey. 1989. Human-computer interaction using eye-gaze input. *IEEE Transactions on systems, man, and cybernetics* 19, 6 (1989), 1527–1534.

[24] Alexander Ivanov, Kurtis Thorvald Danyluk, and Wesley Willett. 2018. Exploration & Anthropomorphism in Immersive Unit Visualizations. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3170427.3188544

[25] Robert J. K. Jacob. 1995. *Eye Tracking in Advanced Interface Design.* Oxford University Press, Inc., USA, 258–288.

[26] Sebastian Kapp, Michael Barz, Sergey Mukhametov, Daniel Sonntag, and Jochen Kuhn. 2021. ARETT: augmented reality eye tracking toolkit for head mounted displays. *Sensors* 21, 6 (2021), 2234.

[27] Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing: Adjunct publication*. 1151–1160.

[28] Arie E Kaufman, Amit Bandopadhay, and Bernard D Shaviv. 1993. An eye tracking computer user interface. In *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*. IEEE, 120–121.

[29] Chaowanan Khundam. 2015. First person movement control with palm normal and hand gesture interaction in virtual reality. In *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 325–330.

[30] Youngjun Kim, Hannah Kim, and Yong Oock Kim. 2017. Virtual reality and augmented reality in plastic surgery: a review. *Archives of plastic surgery* 44, 03 (2017), 179–187.

[31] Hollis Kool. 2016. The ethics of immersive journalism: A rhetorical analysis of news storytelling with virtual reality technology. *Intersect: The Stanford journal of science, technology, and society* 9, 3 (2016).

[32] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A Lee, and Mark Billinghurst. 2018. Pinpointing: Precise head-and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–14.

[33] Michael F Land and Sophie Furneaux. 1997. The knowledge base of the oculomotor system. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 352, 1358 (1997), 1231–1239.

[34] Lik-Hang Lee and Pan Hui. 2018. Interaction methods for smart glasses: A survey. *IEEE access* 6 (2018), 28712–28732.

[35] SK Lee, William Buxton, and Kenneth C Smith. 1985. A multi-touch three dimensional touch-sensitive tablet. *Acm Sigchi Bulletin* 16, 4 (1985), 21–25.

[36] Yang LI, Jin HUANG, Feng TIAN, Hong-An WANG, and Guo-Zhong DAI. 2019. Gesture interaction in virtual reality. *Virtual Reality & Intelligent Hardware* 1, 1 (2019), 84–112. https://doi.org/10.3724/SP.J.2096-5796.2018.0006

[37] Tica Lin, Rishi Singh, Yalong Yang, Carolina Nobre, Johanna Beyer, Maurice A Smith, and Hanspeter Pfister. 2021. Towards an understanding of situated ar visualization for basketball free-throw training. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[38] Tica Lin, Yalong Yang, Johanna Beyer, and Hanspeter Pfister. 2020. SportsXR–Immersive Analytics in Sports. *arXiv preprint arXiv:2004.08010* (2020).

[39] Roman Lissermann, Jochen Huber, Aristotelis Hadjakos, and Max Mühlhäuser. 2013. Earput: Augmenting behind-the-ear devices for ear-based interaction. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 1323–1328.

[40] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. 2020. Design and evaluation of interactive small multiples data visualisation in immersive spaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 588–597.

[41] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-Air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 63–71. https://doi.org/10.1145/2807442.2807489

[42] Ling Luo, Yulia Gryaditskaya, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. 2020. Towards 3D VR-sketch to 3D shape retrieval. In *2020 International Conference on 3D Vision (3DV)*. IEEE, 81–90.

[43] Mathias N Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Grønbæk, 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.

[44] Katsutoshi Masai, Kai Kunze, and Maki Sugimoto. 2020. Eye-Based Interaction Using Embedded Optical Sensors on an Eyewear Device for Facial Expression Recognition. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/3384657.3384787

[45] Rachel Mintz, Shai Litvak, and Yoav Yair. 2001. 3D-virtual reality in science education: An implication for astronomy teaching. *Journal of Computers in Mathematics and Science Teaching* 20, 3 (2001), 293–305.

[46] Tomer Moscovich. 2006. Multi-touch interaction. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 1775–1778.

[47] Tomer Moscovich. 2007. Principles and applications of multi-touch interaction. (2007).

[48] Masa Ogata, Yuta Sugiura, Yasutoshi Makino, Masahiko Inami, and Michita Imai. 2013. SenSkin: adapting skin as a soft interface. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 539–544.

[49] Anneli Olsen, Albrecht Schmidt, Paul Marshall, and Veronica Sundstedt. 2011. Using Eye Tracking for Interaction. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 741–744. https://doi.org/10.1145/1979742.1979541

[50] Patrick Pessaux, Michele Diana, Luc Soler, Tullio Piardi, Didier Mutter, and Jacques Marescaux. 2015. Towards cybernetic surgery: robotic and augmented reality-assisted liver segmentectomy. *Langenbeck's archives of surgery* 400, 3 (2015), 381–385.

[51] Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdelrahman, 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.

[52] 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.

[53] Robin Piening, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, Sarah Prange, Philippe Schröder, and Florian Alt. 2021. Looking for info: Evaluation of gaze based information retrieval in augmented reality. In *IFIP Conference on Human-Computer Interaction*. Springer, 544–565.

[54] Carmelo Pino and Isaak Kavasidis. 2012. Improving mobile device interaction by eye tracking analysis. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 1199–1202.

[55] 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.

[56] Alexander Plopski, Teresa Hirzle, Nahal Norouzi, Long Qian, Gerd Bruder, and Tobias Langlotz. 2022. The Eye in Extended Reality: A Survey on Gaze Interaction and Eye Tracking in Head-Worn Extended Reality. *ACM Comput. Surv.* 55, 3, Article 53 (mar 2022), 39 pages. https://doi.org/10.1145/3491207

[57] Dawid Polap. 2018. Voice control in mixed reality. In *2018 federated conference on computer science and information systems (FedCSIS)*. IEEE, 497–500.

[58] Christopher Prisille and Marko Ellerbrake. 2020. Virtual reality (VR) and geography education: Potentials of 360°'experiences' in secondary schools. In *Modern approaches to the visualization of landscapes*. Springer, 321–332.

[59] Raycast [n. d.]. *Unity Documentation Version: 2021.3 - Physics.Raycast*. Retrieved November 7, 2022 from https://docs.unity3d.com/ScriptReference/Physics.Raycast.html

[60] Jun Rekimoto. 2001. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Proceedings Fifth International Symposium on Wearable Computers*. IEEE, 21–27.

[61] Kadek Ananta Satriadi, Barrett Ens, Maxime Cordeil, Tobias Czauderna, and Bernhard Jenny. 2020. Maps Around Me: 3D Multiview Layouts in Immersive Spaces. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 201 (nov 2020), 20 pages. https://doi.org/10.1145/3427329

[62] Simon Schenk, Marc Dreiser, Gerhard Rigoll, and Michael Dorr. 2017. GazeEverywhere: enabling gaze-only user interaction on an unmodified desktop PC in everyday scenarios. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3034–3044.

[63] Alexandra Sipatchin, Siegfried Wahl, and Katharina Rifai. 2020. Eye-tracking for low vision with virtual reality (VR): testing status quo usability of the HTC Vive Pro Eye. *bioRxiv* (2020).

[64] Niklas Stein, Diederick C Niehorster, Tamara Watson, Frank Steinicke, Katharina Rifai, Siegfried Wahl, and Markus Lappe. 2021. A comparison of eye tracking latencies among several commercial head-mounted displays. *i-Perception* 12, 1 (2021), 2041669520983338.

[65] Athirah Syamimi, Yiwei Gong, and Ryan Liew. 2020. VR industrial applications—A singapore perspective. *Virtual Reality & Intelligent Hardware* 2, 5 (2020), 409–420.

[66] Radu-Daniel Vatavu and Laura-Bianca Bilius. 2021. GestuRING: A web-based tool for designing gesture input with rings, ring-like, and ring-ready devices. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 710–723.

[67] Vive [n. d.]. *VIVE Sense Eye and Facial Tracking SDK Overview*. Retrieved November 7, 2022 from https://developer.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/overview/

[68] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y Chen. 2015. PalmType: Using palms as keyboards for smart glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 153–160.

[69] Jianghao Xiong, En-Lin Hsiang, Ziqian He, Tao Zhan, and Shin-Tson Wu. 2021. Augmented reality and virtual reality displays: emerging technologies and future perspectives. *Light: Science & Applications* 10, 1 (2021), 1–30.

[70] Yalong Yang, Tim Dwyer, Kim Marriott, Bernhard Jenny, and Sarah Goodwin. 2020. Tilt map: Interactive transitions between Choropleth map, Prism map and Bar chart in immersive environments. *IEEE Transactions on Visualization and Computer Graphics* 27, 12 (2020), 4507–4519.

[71] Sang Ho Yoon, Ke Huo, Vinh P Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-worn textile input device with multimodal sensing in mobile interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. 269–272.

[72] Cik Suhaimi Yusof, Huidong Bai, Mark Billinghurst, and Mohd Shahrizal Sunar. 2016. A review of 3D gesture interaction for handheld augmented reality. *Jurnal Teknologi* 78, 2-2 (2016).