Eye-Based Interaction for Achieving Hands-Free Head-Mounted Displays

Submitted to
Graduate School of Information Science and Technology
Osaka University

July 2020

Chang LIU

Thesis Committee:

Prof. Takao Onoye (Osaka University)
Assoc. Prof. Yuki Uranishi (Osaka University)
Prof. Kiyoshi Kiyokawa (Nara Institute of Science and Technology)
Prof. Haruo Takemura (Osaka University)

List of Publications

Journals

1. C. Liu, A. Plopski, and J. Orlosky. OrthoGaze: Gaze-based Three-dimensional Object Manipulation using Orthogonal Planes. *Computers & Graphics*, Vol. 89, 1–10, 2020.

International Conferences

(Under Review)

1. C. Liu, J. Orlosky, and A. Plopski. Eye Gaze-based Object Rotation for Head-mounted Displays, 2020.

Peer-reviewed Papers

1. C. Liu, A. Plopski, K. Kiyokawa, P. Ratsamee, and J. Orlosky. IntelliPupil: Pupillometric Light Modulation for Optical See-through Head-mounted Displays, *Proceedings ISMAR*, 98–104, Oct. 2018.

Peer-reviewed Posters

1. J. Orlosky, C. Liu, D. Kalkofen, and K. Kiyokawa. Visualization-guided Attention Direction in Dynamic Control Tasks, *Proceedings ISMAR*, Oct. 2019.

Domestic Conferences

Non-peer-reviewed Papers

1. C. Liu, A. Plopski, T. Mashita, Y. Kuroda, K. Kiyokawa, and H. Takemura. Automated Backlight Adjustment of Optical See-through HMDs for Improved Visibility, *IEICE technical report*, 115(494):217–222, Jan. 2016. (in Japanese)

Theses

- C. Liu. Automated Backlight Adjustment of Optical See-through Head Mounted Displays for Improved Visibility, *Master's Thesis, Graduate School of Information Science and Technology, Osaka University*, Feb, 2017. (in Japanese)

Abstract

Common head-mounted displays (HMDs) often require users to hold a controller and perform noticeable movements for interaction. In some practical use cases, however, it is difficult to perform such interaction, for example in a crowded public space, when the hands are in use or for physically handicapped users. Though some methods exist for supporting hands-free interaction, e.g. performing gestures or making sound, they usually require noticeable activities, and thus is not completely hands-free or lack social acceptability. To tackle this issue, this work focuses on the development and application of eye-based user interfaces to HMDs for hands-free usability and improved user experience. The focus can be distinguished into two aspects: a) pure hands-free object manipulation of 6 degrees of freedom (DoF), and b) using eye data to improve the functionality of HMDs.

In virtual reality (VR) and augmented reality (AR), gaze-based methods have been explored for decades as effective user interfaces for hands-free interaction. While many methods use eye gaze to assist with hand-based manipulations, interfaces cannot yet provide completely gaze-based 6 degrees-of-freedom (DoF) manipulations in an efficient manner. In this work, a novel user interface, referred to as OrthoGaze, is introduced. OrthoGaze allows the user to intuitively manipulate the three-dimensional position of a virtual object using only eye or head gaze. This approach makes use of three selectable, orthogonal planes, where each plane not only helps guide the user's gaze in an arbitrary virtual space, but also allows for 2-DoF manipulations of object position. To evaluate the method, two user studies were conducted involving aiming and docking tasks in VR to evaluate the fundamental characteristics of sustained gaze aiming and to determine which type of gaze-based control performs best when combined with OrthoGaze. Results showed that eye gaze was significantly more accurate than head gaze for sustained aiming. Additionally, eye and head gaze-based control for 3D manipulations achieved 78% and 96% performance, respectively, in comparison with a hand-held controller. Subjective results also suggest that pure eye gaze-based manipulation can comprehensively cause more fatigue than head gaze-based one. From the experimental results, OrthoGaze is expected to become an effective method for pure hands-free object manipulation in head-mounted displays.

Additionally, three methods are implemented and tested to handle rotations of virtual objects using gaze, including RotBar: a method that maps line-of-sight eye gaze onto per-axis rotations, RotPlane: a method that makes use of orthogonal planes to achieve per-axis angular rotations, and RotBall: a method that combines a traditional arcball with an external ring to handle

user-perspective roll manipulations. The efficiency of each method is evaluated by conducting a user study involving a series of orientation tasks along different axes with each method. Experimental results showed that users could accomplish single-axis orientation tasks with RotBar and RotPlane significantly faster and more accurate than RotBall. On the other hand for multi-axis orientation tasks, RotBall significantly outperformed RotBar and RotPlane in terms of speed and accuracy. In addition, all three methods effectively achieved over 70% matching with the misalignment less than 3 degrees for single-axis tasks, and 70% matching of misalignments less than 6 degrees for multi-axis tasks.

In practical use of optical see-through head-mounted displays (OST-HMDs), users often have to manually adjust the brightness of virtual content to ensure that it is at the optimal level. Automatic adjustment is still a challenge, largely due to the complexity of real world lighting and user perception. As a step towards overcoming this issue, a novel method, referrd to as IntelliPupil, is introduced. IntelliPupil uses eye tracking to properly modulate augmentation lighting for a variety of lighting conditions and real scenes. The system first takes data from a small form factor light sensor and changes in the pupil diameter from an eye tracking camera as passive inputs. The data is coupled with user-controlled brightness selections, allowing the algorithm to fit a brightness model to the user preference using a feed-forward neural network. Using a small amount of training data, both the scene luminance and the pupil size are used as inputs into the neural network, which can then automatically adjust to a user's personal brightness preference in real time. Experiments in a high dynamic range AR scenario with varied lighting show that pupil size is just as important as environment light for optimizing brightness and that the system outperforms linear models.

The results of this work give implication and insights on the design and applications of eye-based user interfaces for HMDs. The author believes that the application of eye tracking technology can lead to great improvement of human-HMD interaction.

Acknowledgments

The writing of this dissertation is an incredible journey and a milestone in my life. I could not have accomplished this endeavor without the tones of support and help from my advisors, friends and family. I would like to give my sincere regards to all of those who supported me during the completion of this work.

This work is done under the supervision of Prof. Haruo Takemura of the Graduate School of Information Science and Technology at Osaka University. I would like to thank him for providing me with continued support and the ideal environment for accomplishing the entire work. I also want to thank the thesis committee for offering me great comments to improve this dissertation.

I want to especially express my sincere gratitude to my supervisor, Assoc. Prof. Jason Orlosky at Osaka University, for his inspiration, assistance and dedication to this work. Throughout my research, he provided me with inspiring ideas, sound opinions, academic experience and encouragement.

I am also grateful to Dr. Alexander Plopski at University of Otago for continuously advising me since the first time I stepped into this field. We had constructive discussions with many great ideas, and shared unforgettable time of co-authoring.

I give my special thanks to my former supervisor, Prof. Kiyoshi Koyokawa at Nara Institute of Science and Technology, who gave me the opportunity to start my research at Osaka University and guide me into this reseach field, which made this current work possible.

I also appreciate the opportunity to be able to study and work with a lot of people at Osaka University. I want to thank all the members of Takemura Laboratory for their support. I would like to especially mention and thank my schoolmate Yuki Tamura for his guidance and help since I first came to the lab. I also want to thank Tao Tao for helping with the photographing for this dissertation. Furthermore, I express my gratitude to the staff at Cybermedia Center and the Graduate School of Information Science and Technology for their help with various official procedures and paperwork.

Lastly and most importantly, I give my sincere thanks to my parents, Jimin Liu and Jianzhen Li, for their understanding and sustained support in my whole life. I could not have had this amazing opportunity to study in Japan and pursue my own interests without their selfless effort. I dedicate this dissertation to them.

Chang Liu
Osaka University
July 2020

Contents

List of Tables

List of Figures

List of Acronyms

ANOVA	Analysis of variance
AR	Augmented reality
CAD	Computer-aided design
CPU	Central processing unit
DoF	Degrees of freedom
FoV	Field of view
FPS	Frames per second
GPU	Graphics processing unit
HCI	Human-computer interaction
HDR	High dynamic range
HMD	Head-mounted display
MSE	Mean square error
OST-HMD	Optical see-through head-mounted display
PC	Personal computer
SUS	System usability scale
VR	Virtual reality

Chapter 1
Introduction

Virtual reality (VR) and augmented reality (AR) technologies have been researched for decades and are now booming in popularity. With the rapid improvement of processors and cameras, users can now easily reach VR and AR applications even through portable equipment such as smart phones with Google Cardboard. As the investment in VR and AR continues to rise, more and more applications have been released to attract consumers in various fields. For example, Sephora Virtual Artist allows users to try virtual makeovers from various real brands on their own photographs. Wayfair, an online furniture shop, is providing an AR-based shopping experience where customers can check the looking of the furniture in their own houses through the camera of the smart phones. As the predicted market size of VR and AR for different use cases can reach a value of $80 billion a year by 2025 (Bellini et al. (2016)), it is promising that the VR and AR have the potential to become the next large computing platform in the future.

VR once showed a similar but temporary boom back in the 1990s when 3D video games, such as Nintendo Virtual Boy and Virtuality, were introduced by gaming companies. The boom eventually faded out because of the poor graphics at the time and expensive prices. Nowadays, as consumer-oriented computers are powerful enough to render virtual content of highly realistic graphics, it seems that the VR and AR don't suffer from the graphic quality and prices that much as in the past. However, when it comes to the spread of VR and AR, there still exist huge obstacles to overcome, among which the user experience is considered as the biggest obstacle to mass adoption of both VR and AR technologies (Karl et al. (2019)). Thus, it is essential to develop novel approaches leading to the improvement of user experience.

The unsatisfying user experience of VR and AR can result from various perspectives such as heavy and bulky hardware, complicated initialization and calibration, and unfriendly interfaces etc. In many situations, excessive user effort is considered as a huge load that needs to be reduced. As such, this work contributes to the improvement of user experience of both VR and AR by developing hands-free interaction methods that are compatible for modern VR and AR devices.

1.1 Background

This section gives a brief introduction on head-mounted displays (HMDs) with the interaction methods, as well as the eye tracking technology, followed up with a discussion on the advantages of utilize eye tracking as a tool for human-computer interaction (HCI) in VR and AR.

1.1.1 Head-mounted Displays

Back in 1935, a short science fiction story titled *Pygmalion's Spectacles* was presented by Stanley G. Weinbaum, where a professor invented a pair of goggles which enables a movie that provides sight, sound, taste, smell and touch. This story describes a comprehensive concept of VR, coupled with the goggle-like display. In 1960, Morton Heilig presented Telesphere Mask which is considered as the world's first HMD. Telesphere Mask provided stereoscopic images along with wide vision and stereo sound, but without any tracking and sensing. Through decades of improvement and evolution, today more and more HMDs with wider field of views and various sensors have been released targeting VR and AR.

Categories. Based on its visual system, an HMD can be generally categorized as an immersive HMD or an optical see-through HMD (OST-HMD), of which the sample is shown in Fig. 1.1. An immersive HMD usually covers the whole view of the user in order to provide a fully immersive VR experience. Thus, the user can only see the generated virtual content and will not be able to view the real world directly. On the other hand, an OST-HMD often uses optical combiners, such as half mirrors, to optically combine the virtual content with the real world directly in the user's view. As a result, the display quality is highly dependent on its optical system. Usually OST-HMDs would have smaller field of views in comparison with the immersive HMDs due to the technological limitations. Table 1.1 shows some main specifications of several latest HMDs.

Interaction Methods. The way of interacting with a device is a determinate factor on its user experience. For basic adjustments of the hardware such as power-on or audio modulation, the most popular way is to use physical buttons, e.g. buttons (Microsoft HoloLens, HTC Vive etc.) or touchpads (Google Glass) on the HMD. At the same time, the essence of interacting with the virtual environment in an HMD relies more on exploiting users' natural abilities rather than learned skills. While interacting with a conventional

Figure 1.1: Sample images of immersive HMDs and OST-HMDs. Left: A user wearing an immersive HMD which provides the user with a fully-immersive VR experience and hence prevents the user from directly viewing the real world. Right: A user wearing an OST-HMD which renders digital content directly in the user's field of view overlying the real world.

computer system requires a set of learned skills, e.g. typing with a keyboard or clicking with a mouse, navigating through a virtual environment exploits users' intuitive activities, e.g. walking with feet or reaching out one's hands to grab something. Accordingly, common interaction methods for HMDs rely upon users' body movements.

Motion Controllers. As listed in Table 1.1, motion controllers have become a major interaction tool for the immersive HMDs. Controllers work as an extension of users' arms for directly interacting with virtual objects nearby, and can be coupled with other methods, e.g. spotlights (Liang and Green (1994)) , raycasting (Mine (1996)), or scaling of the virtual world (Mine et al. (1997)), for interaction out of reach. For controller-based interaction, users

Table 1.1: Specifications of various HMDs.

Device	Resolution	FoV [°]	Wireless	Interaction	Eye Tracking
a. Immersive					
HTC Vive Cosmos	$1440 \times 1700 \times 2$	110	no	motion controller	no
HP Reverb	$2160 \times 2160 \times 2$	114	no	motion controller	no
HTC Vive Pro Eye	$1440 \times 1600 \times 2$	110	no	motion controller	yes
Oculus Rift S	$1280 \times 1440 \times 2$	115	no	motion controller	no
Oculus Quest	$1440 \times 1600 \times 2$	110	yes	motion controller	no
b. Optical see-through					
Microsoft HoloLens 2	2K	−−	yes	hand gesture, voice	yes
Magic Leap One	1280×960	40	yes	hand-held pad	no
MOVERIO BT-300	1280×720	23	no	hand-held pad	no

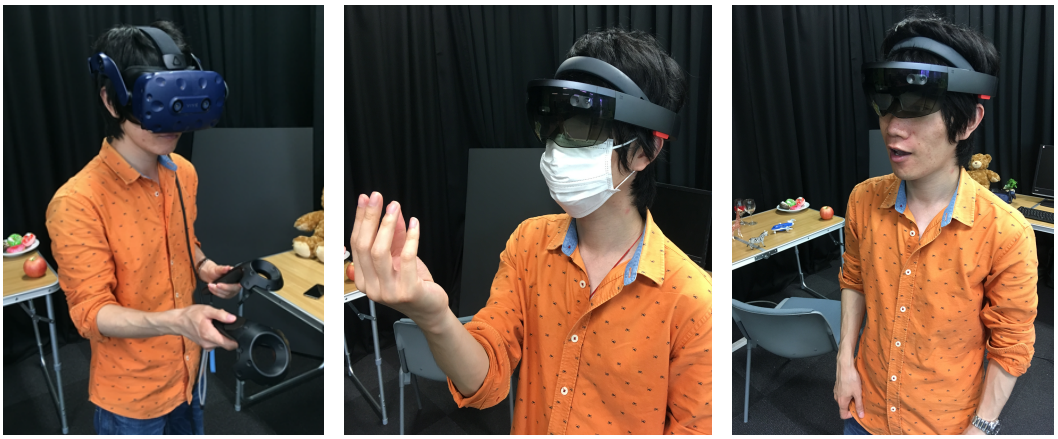


Figure 1.2: Images showing various interaction methods commonly used for HMDs. Left: Hand-held controllers used for immersive HMDs. Center: Specific hand gestures utilized for interacting with Microsoft HoloLens. Right: Talking with the voice interface (Cortana) on Microsoft HoloLens. All of the three methods are noticeable from an observer's perspective.

will often have to handhold one physical controller or a pair of controllers. The controllers usually have integrated motion sensors such as gyroscopes and accelerometers to track their basic movements. Additionally, to achieve spatial tracking, controller-based interaction often requires other tracking devices such as mounted cameras (inside-out methods) or fixed base stations (Lighthouse). In general, the controller-based interaction has good capability of interacting with VR content intuitively and accurately since it can reach a 6-degrees-of-freedom (6-DoF) manipulation in a relatively easy way comparing with other methods.

Body Gestures. Besides motion controllers, body gestures, especially hand gestures, have also become a basic interaction method. At the present day, it is possible to track users' hand motion as well as the fingertip movements without the requirement for tracking visually distinct markers (Lee and Hollerer (2007)), which has already been applied in industrial products. For instance for Microsoft HoloLens, users can perform two basic hand gestures, the Air Tap and the Bloom, to manipulate the digital content. In comparison with controllers, gesture-based interaction frees users' hands as it is not physically occupying hands, which is more user-friendly in case of OST-HMDs. Also using one's sense of position and orientation of his/her body and its several parts, which is known as the proprioception (Boff et al. (1986)), can helps with improving the intuitiveness of the interaction as well (Mine et al.

(1997)).

Voice. While controllers and gestures can effectively help with the human-HMD interaction, the requirements of excessive hardware and extensive body movements still stand as huge obstacles causing highly fatiguing user experience. One approach to reducing the physical fatigue is to use voice as an interaction tool. Voice user interfaces (VUIs) have been widely embedded into our everyday life through smartphones and some assistant devices, e.g. Siri and Cortana. A common advantage of a VUI is that it is simple and efficient for users to make specified orders to the device, such as launching software and searching for targets.

1.1.2 Problem Definition

In the world of science fiction literature, HMDs are often depicted as light, easy-to-wear and intelligent devices, such as glasses or goggles that can support the user with necessary information, in arbitrary occasions and with minimal interaction. This can be considered as one of the ultimate goals for future HMDs. At the present day, however, HMDs are still far away from such goal. HMDs are usually massive and requires extensive effort for interaction. While the mass can be reduced along with the evolution of manufacture, the improvement in interaction asks for more of inspiration in its methods.

One primary limitation for the popular controller-based interaction is that the specified controllers occupy users' hands and thus limit the hands' fundamental functions, e.g. users cannot hold the controllers and a cup of coffee at the same time. Besides, in some practical cases it is difficult to use controllers, e.g. in a crowded train or for physically handicapped users. Gesture-based interaction also suffers from such limitations. Thus hands-free interaction methods requiring lower physical effort are needed.

Though VUIs exist as a straightforward method for hands-free HCI, it is difficult to perform motion-based movements through VUIs, such as positioning objects. Besides, a VUI usually requires users to make sound (voice), which has the same issue as gestures as it could become socially embarrassing in some use cases. Thus, other hands-free methods of higher social acceptability need to be developed.

To tackle this issue, this work proposes eye-based interaction for improved usability and user experience of HMDs.

1.1.3 Eye Tracking

Eyes are one of the most salient features on the human face. As part of the sensory nervous system, eyes provide the brain with optical information for visual cognition of the outside world. Compared with other organs such as the nose or ears, the action of eyes themselves are very active and noticeable. For instance daily eye activities such as gaze movements or blinks can be easily recognized from a third person's perspective. There also exist some activities that are subtle but still can be captured, e.g. changes in pupil size affected by incoming light or undulating emotions.

Eye tracking can be facilitated by different methods that either use the iris contour (Wu et al. (2007)), the pupil contour (Kassner et al. (2014)), or a combination of iris/pupil contours with reflections of the environment in the eye (Plopski et al. (2015); Guestrin and Eizenman (2006)). Krafka et al. (2016) also proposed a eye tracking model trained by a convolutional neural network fed with a large-scale dataset which can provide accurate eye tracking for modern mobile devices. Through eye tracking, various information is available regarding the features of the eye and its activities, e.g. eye gaze, eye pose, pupil response, iris patterns and corneal reflections etc. (Fuhl et al. (2016); Nakazawa and Nitschke (2012)). Additionally, information provided by eye tracking also plays an essential role in understanding and interpretation of adaptation, attention and cognition states.

As researchers' interests in eye activities rose in the early 1900s, initial eye tracking technologies were developed to track eye movements in reading (Huey (1908)). Throughout years of research and development, the use of eye tracking has expanded to numerous fields, providing tremendous insights to various domains such as human vision and perception, education, medical science and neuroscience (Duchowski (2002)). Recently, thanks to the rise in the performance of computers and cameras, eye tracking has also been applied to the development of novel user interfaces in the domain of HCI.

Eye Gaze-based User Interfaces. When users are interested in an object they tend to look at that object. This tendency means that eye gaze can be a natural modality for interaction with virtual content. Using the eye as an interaction tool has several advantages. Firstly, the eye gaze can work as a fast on-screen pointing cursor. Ware and Mikaelian (1986) observed that the eye is faster than a mouse for performing pointing movements. Secondly, the eye-tracking-based operation itself is easy to perform. Though sometimes it takes time to get accustomed to a certain system, an eye-tracking-based user interface can usually be actuated by basic eye activities such as gaze movements

and dwelling. Therefore, unlike gesture-based user interfaces, less training is needed for eye-based ones. Moreover, eye tracking is suitable for developing hands-free user interfaces. Hands-free interaction is rather important for HMDs considering practical use cases. It not only helps with improving the usability when hands are occupied by other tasks, but also has the potential to raise the social acceptability of HMDs, as eye activities are much less noticeable from a third perspective compared to other interaction methods such as gestures or voice.

Midas Touch Problem. Although gaze is a natural modality, it suffers from several issues such as poor stability and unintentional selections, also referred to as the Midas Touch problem (Jacob (1995)). The Midas Touch problem can result from a naive implementation that is without any form of triggering to confirm the further interaction. For example, an interface of simply showing detailed information of the item that the user's eye gaze engages can cause a circumstance that the information is displayed for everything the user looks at despite whether he/she wants or not.

As the eye gaze is an extremely natural activity in daily life, for most of the time users are not expecting a simple "look at" to actually "mean" something. This makes the Midas Touch problem a critical issue for eye-based user interface design, as a system with the Midas Touch problem can annoy the user with unwanted responses and as a result prevent the user from performing further engagements. Thus a friendly-designed eye-based user interface should process the passive eye tracking inputs carefully in order to avoid the Midas Touch problem.

Over the years researchers have developed different methods to overcome the Midas Touch problem. A common solution to this problem is to utilize a dwell timer that requires users to focus on the target object for a given time period to differentiate between intentional focus and exploring gaze (Jacob (1990)). Alternatively, users can confirm the selection through button clicks (Jalaliniya et al. (2015)), speech (Beach et al. (1998)), and gaze gestures (Istance et al. (2008); Kytö et al. (2018); Vidal et al. (2013a); Khamis et al. (2018)).

1.1.4 Advantages of Eye Gaze-based Interaction

For human-HMD interaction, the eye gaze-based methods are considered to be able to improve the user experience primarily from two perspectives.

Low Manual Interaction Requirements. By using eye movements, it is expected to effectively reduce the effort for interaction from the user's perspective. In comparison with traditional methods such as hand-held controllers or gestures that require extensive body movements, making eye movements requires less muscle effort, and hence can reduce the fatigue through long-term use. Moreover, eye-based interaction can reduce, or completely cut down the requirement of using hands. Freeing the hands enables users to work with the HMD in a more efficient way, which meets the requirement of developing HMDs that are without necessity of extensive hardware and can be used in arbitrary occasions.

High Social Acceptability. You may not feel comfortable to use the HMD in the street if other passengers are looking at you as if they are looking at a weird person. That is to say, the social acceptability is also an important factor to the user experience of HMDs. Compared to gestures or VUIs, eye-based interaction is much less noticeable from a third perspective, which contributes to the improvement of social acceptability of HMDs, since it is observed that for human-HMD interaction, inputs that are less noticeable are more socially acceptable from perspectives of both users and observers (Alallah et al. (2018)).

The next section identifies particular issues this work is tackling regarding human-HMD interaction, and summarizes the approaches to those issues along with the main contributions.

1.2 Contributions

This work proposes eye-based methods to human-HMD interaction for improving the user experience of HMDs. In this work, applying eye tracking is subsequently composed of two subordinate aspects: 1) to provide pure hands-free manipulation of high degrees of freedom and 2) to automating functionality and reduce manual effort for using HMDs.

Gaze-based Three-dimensional Object Positioning. In VR and AR, gaze-based methods have been explored for decades as effective user interfaces for hands-free interaction. Though several well-known gaze-based methods exist for simple interactions such as selection, no solutions exist for 3D manipulation tasks requiring higher degrees of freedom (DoF). This work introduces a novel user interface, referred to as OrthoGaze, that allows the user to intuitively manipulate the three-dimensional position of a virtual object

using only the eye or head gaze. This approach makes use of three selectable, orthogonal planes, where each plane not only helps guide the user's gaze in an arbitrary virtual space, but also allows for 2-DoF manipulations of object position.

To evaluate the method, this work conducted two user studies involving aiming and docking tasks in virtual reality to evaluate the fundamental characteristics of sustained gaze aiming and to determine which type of gaze-based control performs best when combined with OrthoGaze. Results showed that eye gaze was more accurate than head gaze for sustained aiming. Additionally, eye and head gaze-based control for 3D manipulations achieved 78% and 96% performance, respectively, in comparison with a hand-held controller. Subjective results also suggest that pure eye gaze-based manipulation can comprehensively cause more fatigue than head gaze-based one. From the experimental results, OrthoGaze is expected to become an effective method for pure hands-free object manipulation in head-mounted displays. In summary, the contributions of this work are:

- This work presents a novel approach that enables hands-free adjustment of virtual object position in HMDs.
- An experiment is conducted that evaluates sustained eye-gaze and head-gaze aiming in a painting task. The results show that eye-gaze outperforms head-gaze in terms of accuracy. Furthermore, in some cases larger areas can be covered with eye-gaze than head-gaze.
- The experimental results show that for 3D docking tasks, eye and head gaze-based control with OrthoGaze can achieve 78% and 96% success rates, respectively, when compared to a hand-held controller.

Gaze-based Three-dimensional Object Rotation. Hands-free manipulation of 3D objects has long been a challenge for VR and AR. While many methods use eye gaze to assist with hand-based manipulations, interfaces cannot yet provide completely gaze-based 6-DoF manipulations in an efficient manner.

To address this problem, this work explored three methods to handle rotations of virtual objects using gaze, including RotBar: a method that maps line-of-sight eye gaze onto per-axis rotations, RotPlane: a method that makes use of orthogonal planes to achieve per-axis angular rotations, and RotBall: a method that combines a traditional arcball with an external ring to handle user-perspective roll manipulations. This work validated the efficiency of each method by conducting a user study involving a series of orientation tasks

along different axes with each method. Experimental results showed that users could accomplish single-axis orientation tasks with RotBar and RotPlane significantly faster and more accurate than RotBall. On the other hand for multi-axis orientation tasks, RotBall significantly outperformed RotBar and RotPlane in terms of speed and accuracy. In addition, all three methods effectively achieved over 70% matching with the misalignment less than 3 degrees for single-axis tasks, and 70% matching of misalignments less than 6 degrees for multi-axis tasks. In summary, the contributions of this work are:

- Three different methods are implemented that improve upon existing work and redesigned them to specifically address the needs of those who need eye-only control to rotate virtual objects.
- A user study is conducted that compares these methods in different situations to determine their suitability for simple and complex object manipulations. The results show that RotBar and RotPlane are more suitable for simple rotations and RotBall is more suitable for complicated manipulations.
- Results of the user study revealed several observations that need to be considered by future interfaces for gaze-based manipulation.

Automated Light Modulation for OST-HMDs. This work is mainly dedicated to reducing the manual adjustment requirements for using OST-HMDs. In practical use of OST-HMDs, users often have to adjust the brightness of virtual content to ensure that it is at the optimal level. Automatic adjustment is still a challenging problem, largely due to the bidirectional nature of the structure of the human eye, complexity of real world lighting, and user perception. Allowing the right amount of light to pass through to the retina requires a constant balance of incoming light from the real world, additional light from the virtual image, pupil contraction, and feedback from the user. While some automatic light adjustment methods exist, none have completely tackled this complex input-output system.

As a step towards overcoming this issue, this work introduces IntelliPupil, an approach that uses eye tracking to properly modulate augmentation lighting for a variety of lighting conditions and real scenes. The system first takes data from a small form factor light sensor and changes in the pupil diameter from an eye tracking camera as passive inputs. The data is coupled with user-controlled brightness selections, allowing the algorithm to fit a brightness model to the user preference using a feed-forward neural network. Using a small amount of training data, both the scene luminance and the pupil size

are used as inputs into the neural network, which can then automatically adjust to a user's personal brightness preference in real time. Experiments in a high dynamic range AR scenario with varied lighting show that pupil size is just as important as environment light for optimizing brightness and that the system outperforms linear models. The primary contributions of this work include:

- This work proposes a novel algorithm for OST-HMDs that accounts for user preference, pupil size, and environment light to automatically manage display brightness of OST-HMDs.
- The experimental results reveal that pupil size is just as important as environment light for optimizing brightness and that IntelliPupil outperforms linear adjustment methods in matching the user preference of the display brightness.

1.3 Dissertation Overview

The remainder of this dissertation is composed as follows:

In Chapter 2, a survey is provided relating to the historical attempts on exploring eye gaze-based user interfaces. The review of existing work is also presented for better understanding and judgement of this dissertation. The necessity and challenges regarding achieving hands-free interaction for HMDs is subsequently introduced as the initial motivation of this work.

Chapter 3 introduces a novel method for achieving hands-free object manipulation for HMDs. An orthogonal-plane design is used to allow for pure hands-free gaze-based manipulation. A comprehensive set of experiments is conducted to verify the efficiency of this method in comparison with controller-based manipulation. A summary discusses the experimental results along with the implications.

Chapter 4 introduces a work that explores the efficiency and usability of different gaze-based object rotation methods for HMDs. Three different methods are presented and implemented, with a user experiment conducted to reveal the quantitative performance and the qualitative user experience of each method. A summary then discusses the experimental results along with the implications.

Chapter 5 introduces a novel method for achieving hands-free light modulation of OST-HMDs. This method accounts for user's pupil response and uses it as an input to train a customized neural network in order to get a user-calibrated model. A user experiment in a real scene is conducted to con-

firm the effectiveness and the efficiency of the light modulation. A summary discusses the experimental results along with the implications.

Chapter 6 summarizes and concludes this dissertation with a detailed overview of the contributions and findings of this work, and discusses the remaining challenges regarding the future development of hands-free human-HMD interaction.

Chapter 2

Related Work

This chapter gives an introduction on related studies in the domain of eye gaze-based HCI and systems enhanced with eye-based techniques, as well as the further motivation for this work. Since highly related studies will be also introduced in the chapter of each specific work, this chapter mainly summarizes related studies in a broad sense.

2.1 Eye Gaze-based Interaction

Eye gaze has been proposed as an input paradigm for a number of applications. For example, Majaranta and Räihä (2007) proposed that text input via eye gaze is possible. Their experimental results also show that small improvements in the interface design, e.g. adding a simple "click" that confirms the selection by gaze, can lead to significant improvements in user experience and satisfaction. Smith et al. (2005) explored the usage of eye gaze dwell as a interaction means towards ubiquitous computers embedded in real objects. Gaze has also been utilized for control of physical tools and robotics. For example, Ktena et al. (2015) used sequential eye movements as an input for hands-free wheelchair control and training. Theofilis et al. (2016) used gaze for the remote teleoperation and viewing of an surrogate robot.

Eye gaze is also useful for supporting a number of object selection and manipulation tasks (Johansson et al. (2001)). In most cases however, eye gaze is used in combination with other input modalities. For example, Pouke et al. (2012) introduced a system using gaze and hand gestures targeting object manipulation on tablet devices. Stellmach and Dachselt (2013) proposed a system that allows users to seamlessly manipulate 2D objects on smartphones using a combination of head/eye gaze and screen touch.

While all these eye interactions are available, eye gaze has also been widely explored as a means of interaction in the domain of VR and AR.

2.1.1 Applications in VR and AR

In VR and AR, eye gaze can also be used as a means for automating certain functions and for supporting object manipulation and selections.

Content Arrangement. It has been found that eye tracking can help with content arrangement in VR and AR for improved user experience. Toyama et al. (2015) introduced that gaze and vergence can be used for automated control of dimming and brightening the screen to assist users with switching from screen content to scene content (Fig. 2.1 Left). The system automatically judges whether a user is engaged with virtual content in the display or focusing on the real environment and then determines his or her cognitive state. Based on these analytic capacities, several proactive system functions are implemented including adaptive brightness, scrolling, messaging, notification, and highlighting, which would otherwise require manual interaction. The experimental results show robustness of the attention engagement and cognitive state analysis methods. A majority of the participants (8/12) stated the proactive system functions are beneficial.

Similarly, McNamara et al. (2018) proposed the use of gaze as a passive input for annotation activation. Experimental results show that integrating eye tracking into VR environments to dictate where and when textual information is presented can improve performance when searching for contextual information.

Object Selection and Manipulation. Eye gaze has been promoted as a means for object selection in VR and AR. Tanriverdi and Jacob (2000) proposed a system that uses eye gaze as a tool for selection in VR environments. They found through experiments that selection with eye gaze was significantly faster than with hand pointing, especially for selecting distant objects. As such, it is expected that the eye gaze could serve as a means to perform fast interaction with interfaces in VR.

More recently, eye gaze has also been explored as a collaborative tool for supporting object manipulation. For example, Song et al. (2014) proposed GaFinC, a multi-modal method using finger and gaze for 3D manipulation targeting computer-aided design (CAD). In GaFinC, the eye gaze is used as the point of interest and has no function on actual manipulation. Pfeuffer et al. (2017) introduced a comprehensive system where eye gaze is combined with finger pinch gestures to achieve 3D manipulation in an immersive space. In these systems, the user could firstly utilize eye gaze to select virtual objects and then has to grab or manipulate them with hand gestures.

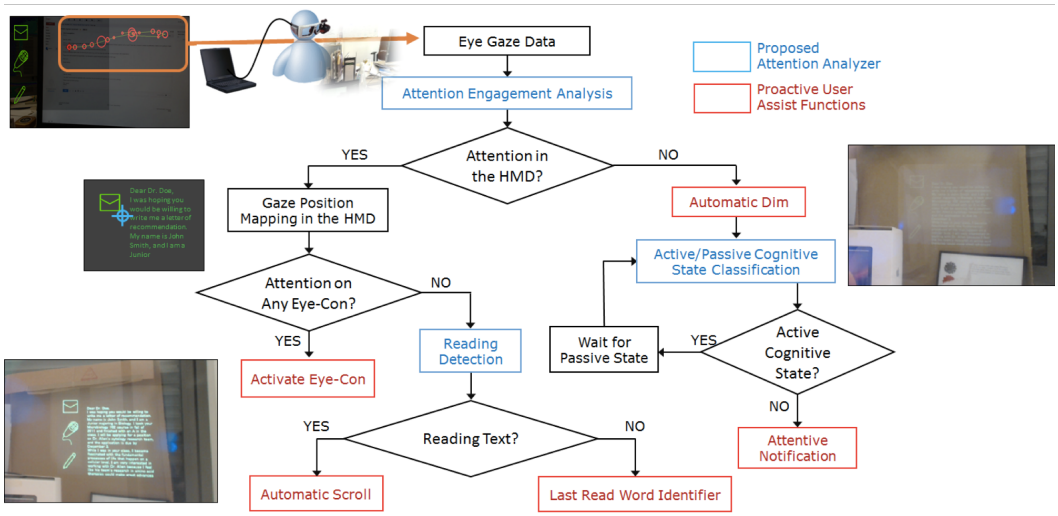


Figure 2.1: Images showing gaze-based content arrangement method for OST-HMDs (Toyama et al. (2015)). The entire interaction is driven by attention engagement analysis and cognitive state analysis, and is implemented with various proactive assist functions such as automatic dim, automatic text scroll and attentive notification.

By including eye gaze, eye-hand collaborative systems benefit to a large extent from the advantage of quick pointing afforded by eye selection. However, while many systems managed to use eye gaze to assist with hand-based manipulations, interfaces cannot yet provide completely hands-free 6-DoF manipulation in an efficient manner. Achieving completely gaze-based manipulations asks for large-scaled prototyping and user studies, which still remains unexplored in the domain.

2.1.2 Addressing the Midas Touch Problem

As introduced in Sec. 1.1.3, simply implemented eye gaze-based user interfaces can lead to unintended triggering by users, which is known as the Midas Touch Problem. Since it can severely impact the user experience, the Midas Touch Problem is a critical factor for researchers and developers to pay attention to while designing eye gaze-based user interfaces. Over decades various methods have been explored to tackle the issue.

Pursuit-based Approaches. Drewes and Schmidt (2007) suggested that using complex eye gestures as an input modality can effectively avoid the Midas Touch problem. Following the same concept, Vidal et al. (2013a) proposed Pursuits, where the system makes a response only when it judges that the user

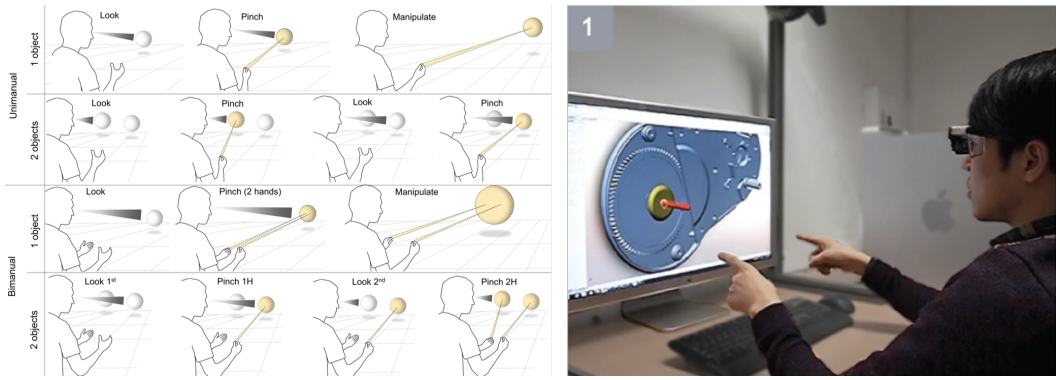


Figure 2.2: Images showing gaze-supported object manipulation in VR and AR. Left: A user can use eye gaze to target virtual objects and use finger gestures for further manipulation (Pfeuffer et al. (2017)). Right: A gaze-supported CAD system where eye gaze is used as the point of interest, coupled with finger-based manipulation (Song et al. (2014)).

is pursuing a specific moving trajectory displayed on the monitor (Fig. 2.3 Left). Given the difference between gaze movements and moving trajectories, the Pursuits algorithm is able to detect specific eye movements and allows for gaze-based selections. The experimental results showed that the Midas Touch Problem can be effectively avoided, and that users can interact with pursuit-based interfaces without prior knowledge or preparation phase.

Follow-up research by Khamis et al. (2018) showed that pursuit-based interfaces are compatible of HCI in the VR environment. Mattusch et al. (2018) also found that users can still select targets quickly via Pursuits even if their trajectory is up to 50% hidden, and at the expense of longer selection times when the hidden portion is larger.

Dual-gaze-based Approaches. Mohan et al. (2018) proposed DualGaze to avoid the Midas Touch Problem when selecting objects by applying a distinctive two-step confirmation. Once users gaze upon a selectable object, a confirmation flag pops up next to the object at a location where the users' gaze just passed through. This trajectory-adaptive flag placement strategy reduces the chance of unintended triggering by requiring a returning gaze back to the flag. The experimental results showed that the gual-gaze-based interface works efficiently enough for selection tasks in VR environments and can help reduce the unintended selection made by users.

Piumsomboon et al. (2017) also proposed novel dual-gaze-based, pursuit-based and head gaze-based user interfaces in VR environments. The interfaces are evaluated compared with the gaze-dwell-based method. Their experimen-

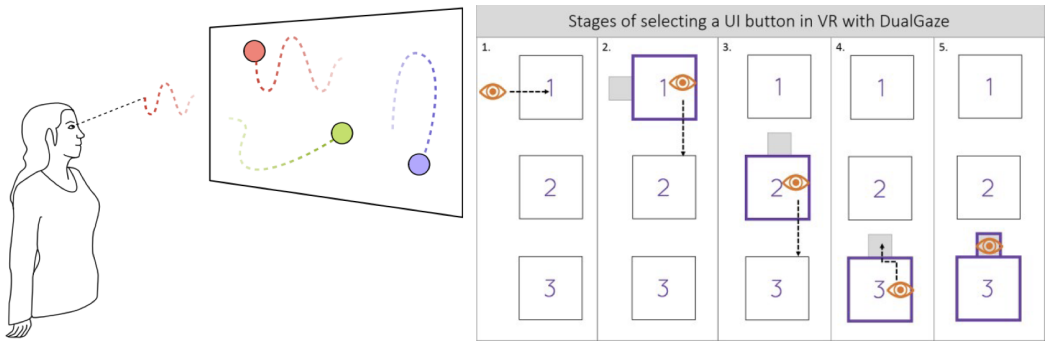


Figure 2.3: Images showing the Pursuits and DualGaze methods. Left: Pursuits matches user's smooth eye pursuits with on-screen moving objects (Vidal et al. (2013a)). Right: The stages of the DualGaze method, with the red eye denoting the user's gaze point, involving a user's gaze travelling through buttons 1 and 2 to confirm the user's desired choice, button 3. The confirmation flag is positioned inconsistently from the gaze trajectory and thus forces the user to gaze back to confirm (Mohan et al. (2018)).

tal results showed that the three methods above have similar performance as the gaze-dwell-based method but have superior user experience.

While both pursuit-based and dual-gaze-based methods can effectively help with avoiding the Midas Touch Problem, however, they are designed for selection tasks and none of them has achieved interaction of higher DoF.

2.2 Systems Enhanced with Eye-based Techniques

Eye movements have the ability to provide insights into visual, cognitive and attentional aspects of human performance that are essential to the study of human factors. As such, the analysis of eye movements can also serve as a powerful additional mechanism for measuring human factors to improve system functionality in particular.

For example, eye tracking has been used for decades as an analytical tool of human factors in simulators. Anders (2001) introduced a sophisticated flight simulator certified for airline training. In their simulator, eye movements contributed by providing the analysis of pilot attention allocation, where they could specify the areas of interest within the primary flight display (which is a main information source to the aircraft pilot in the cockpit). Graeber and Andre (1999) examined pilots' eye movements to understand visual attention while using electronic moving maps in different visibility conditions. Ho et al. (2001) measured drivers' eye movements and fixations for analyzing visual

search of traffic signs in high-clutter driving scenes.

While analysis of eye movements can provide important human factor information for using particular systems, it can consequently also be included to enhance the system. Liu (1998) explored the possibility of analyzing eye movements for understanding drivers' mental processes to improve driving comfort and safety. They showed that precise eye movement analysis is able to enhance the system's performance by enabling recognition of driver intentions. Vertegaal (1999) introduced a gaze-aided multiparty communication system. The system benefits from visualized eye gaze direction of each attendee indicating "who is talking to whom, and who is talking about what." More recently, eye trackers have become available for displaying the eye gaze position of the online live streamers, usually when they are gaming [1]. As such, the audience can clearly understand in real-time what exact area on the monitor the streamer is focusing on.

Although eye movements have been found useful for improving the functionality of some systems, it is difficult to directly apply those insights to the case of HMDs due to the different modalities. In addition to eye movements, it is also expected that there exist other potential information accessible from eye tracking that can enhance the functionality of HMDs. As eye trackers are becoming more viable for HMDs, it is worthwhile to dig deeper in this unexplored domain for novel insights on eye-based factors that could particularly benefit the performance and usability of HMDs.

2.3 Motivation

The work of this dissertation focuses on applying eye gaze-based approaches specifically to human-HMD interaction. The focus falls into two aspects: 1) to achieve pure hands-free human-HMD interaction of a high DoF, and 2) to use eye-based factors to reduce manual adjustment requirements and thus improve the functionality of the HMD.

2.3.1 Pure Hands-free Interaction of High DoF

While the existing gaze-hand collaborative approach can handle the manipulation tasks, most of them use the eye gaze as a supportive tool for target selection followed with the hand gestures for manipulation. Although some 2D interfaces use only eye gaze to select and perform actions (Hornof et al. (2004)), there still lacks thorough work on pure eye gaze-based methods for

[1] https://gaming.tobii.com/software/ghost/

object manipulation of high DoF (e.g. 3-DoF positioning and 3-DoF rotation). Since object manipulation is the basic interaction with virtual objects, it is necessary to explore methods that actually enables pure eye gaze-based 3D manipulation with high performance. In this dissertation, Chapter 3 and Chapter 4 contributes to this topic.

2.3.2 Improving Functionality of HMDs

In practical use cases of HMDs, manual effort is often required for achieving better user experience. For instance, users will need to adjust the brightness of an OST-HMD in order to adapt to the lighting condition of the real scene background. As the real scene condition changes from time to time, the requirement of manual adjustments can become frequent. Such extensive manual adjustment requirement can potentially damage the user experience to a certain extent and thus needs to be reduced. While a real, scene-based method is often used to enable automatic light adjustment, it is not efficient since matching the user's preference to a world sensor's limited dynamic range and the different field of view from the user's actual perception can be difficult. To tackle this issue, Chapter 5 introduces an approach that uses real-time pupil response data for automatic brightness adjustment of OST-HMDs.

Chapter 3

Gaze-based Three-dimensional Object Positioning

3.1 Introduction

In recent years, head-mounted displays (HMDs) have enabled the use of virtual and augmented reality (VR and AR). One persisting issue with content in these devices is lack of the ability to author or manipulate content in a hands-free manner.

Over the years, many different manipulation techniques have been developed that enable direct authoring (Bowman et al. (2004)), such as gesture-based selection and manipulation (Van den Bergh and Van Gool (2011)), the Go-Go technique (Poupyrev et al. (1996)), or avatar representation (Slater et al. (2008); Argelaguet et al. (2016)) to name a few. Although these have been shown to be efficient, they often require additional hardware and tracking capabilities that may not always be available.

Furthermore, practical use cases exist, e.g. in a crowded public space or small room, where the use of body gestures and peripheral objects could disturb others or occlude surroundings. Research also suggests that performing noticeable movements for interacting with a device may reduce its social acceptability (Alallah et al. (2018)). As such, interaction methods that do not attract extensive attention and allow for discrete manipulation of content are essential.

Handheld devices such as joysticks, mobile phones and tablets can help address this problem (Mohr et al. (2019)). Nevertheless, the requirement for peripheral devices makes interaction difficult when the user's hands are preoccupied or when using an additional device is not an option. Very recently, commercial HMDs such as the Microsoft HoloLens 2, MagicLeap One and HTC Vive Pro Eye, have begun to include integrated eye tracking, which has great potential for enabling hands-free interaction.

The interest in eye-based interaction is partly due to the tendency to direct our gaze towards objects we are interested in, which makes it a good indicator of the user's intention and focus (Langton et al. (2000)). Gaze techniques have

been widely studied for use in 2D interfaces as a means of estimating the effects of different interfaces on user focus (Takagi et al. (2001)), selection of items (Colombo and Del Bimbo (1997)), or to design attentive interfaces that react to the user's gaze (Kumar et al. (2007)). While similar applications have emerged on HMDs as well (Toyama et al. (2015)), it is difficult to directly transfer interaction methods from a 2D to a 3D interface due to the higher degrees of freedom.

Furthermore, although one can determine the user's gaze point on the screen, it is much more difficult to measure the depth at which the user is focusing or for users to manipulate the focal depth voluntarily without a reference object (Lee et al. (2017)).

The goals of this work are to address the need for a hands-free manipulation method that is both discrete and can handle higher DoF operations. This work presents OrthoGaze, a novel interface that allows users to manipulate the 3D position of virtual objects using only eye or head gaze. As shown in Fig. 3.1, users are presented with three semi-transparent orthogonal planes that define three different interactive dimensions. Users can choose a plane on which they want to move the object on and subsequently manipulate the position of the object on that plane using gaze-plane intersection, i.e., the intersection of the user's gaze on the active plane. While OrthoGaze is in some ways similar to INSPECT, which allows for 6-DoF control on mobile phones (Katzakis et al. (2015)), it has the advantage of being hands-free, and does not require external hardware.

In the past, some studies have found that head gaze outperforms eye gaze (Qian and Teather (2017)), and others have found contrasting results (Blattgerste et al. (2018)). Furthermore, these papers focused on selection in 2D tasks and have yet to consider manipulation in 3D. However, it is expected that using head gaze for 3D manipulation requires more extensive and accurate head movements than for 2D tasks. Performance can also be affected by constraints of the head's angular motion due to the limitations of neck rotation (Kuo et al. (2018)). As OrthoGaze supports both eye and head gaze, it is necessary to test how well each type of gaze-based control would perform for 3D aiming and manipulation tasks. As such, the first step to evaluate OrthoGaze was to test a user's ability to target different areas on the presented planes to compare how accurately and easily users can adjust the targeting location. The results show that depending on the targeted plane eye gaze can outperform head gaze and in general was more accurate than head gaze.

A second experiment was then conducted to evaluate how well users could reposition a virtual object to a target location at different depths. For this

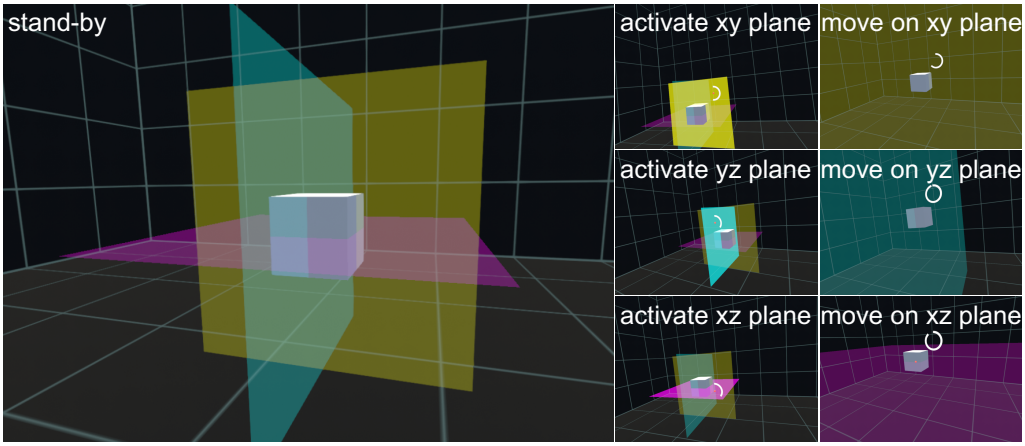


Figure 3.1: OrthoGaze enables gaze-based position manipulation of virtual objects in 3D. (left) The user can move the object on the three orthogonal planes displayed around the object. (middle) When the user wants to select a plane, it is highlighted, the other planes are dimmed out, and a dwell timer indicates the selection. (right) After activating a plane, the user can move the object around to adjust its 2-DoF position by looking at the target location and confirming placement through a gaze dwell.

purpose, this work tested OrthoGaze with eye gaze, head gaze, and raycasting controller for docking tasks as a comprehensive comparison. Though a controller is not hands-free, a controller was chosen with raycast and trigger selection as the baseline to determine how well head and eye gaze-based manipulation could match hand-based performance. As expected a controller with raycast performed the best in both qualitative and quantitative measures, but OrthoGaze still enables efficient 3D manipulation for head and eye gaze as well. At the same time, results were found contrasting from those of experiment one. Participants rated eye gaze-based control lower than head gaze-based control, and were more successful with head gaze than eye gaze.

In summary, contributions of this work are:

- This work presents OrthoGaze, a novel approach that enables hands-free adjustment of virtual object position in HMDs.
- An experiment is conducted that evaluates sustained eye gaze and head gaze aiming on planes. The results show that eye gaze outperforms head gaze in terms of accuracy. Furthermore, in some cases larger areas can be covered with eye gaze than head gaze.
- Results show that for 3D docking tasks, eye and head gaze-based control with OrthoGaze can achieve 78% and 96% success rates, respectively,

when compared to a hand-held controller.

3.2 Related Work

Most of the work related to this research primarily falls into two domains: (a) methods for object position manipulation, and (b) gaze-supported interaction.

3.2.1 Object Position Manipulation

As described by Bowman et al. (2004) selection and manipulation is a basic element of 3D interaction with a large variety of hardware and software solutions to facilitate it. The most natural interaction method is direct manipulation of virtual objects by picking them up with our hands and manipulating their location. The main limitation of this interaction technique is that it is only applicable within our immediate vicinity. One way to address this limitation is the world-in-miniature technique that presents a miniaturized version of the world in front of the user that replicates any adjustments to virtual objects to their counterpart (Stoakley et al. (1995)). Chae et al. (2018) applied the same idea in an AR context where they use a wall for supporting the manipulation of distant virtual objects. Another common technique is to use raycasting methods with a depth-manipulation technique, e.g., Go-Go technique (Poupyrev et al. (1996)).

While these techniques require potentially large hand movements, miniature mice (Nanayakkara et al. (2013)) and handheld devices (Katzakis et al. (2015)) can provide discrete manipulation with virtual content. The most basic form is the use of buttons that adjust the location of the virtual object whenever they are activated or entry fields where the user can adjust the position of the object (Castle and Murray (2009)). While this provides the most control, it is time consuming. Instead of buttons, a virtual object can be moved along displacement vectors of a joystick handle (Simon and Doulis (2004)). INSPECT (Katzakis et al. (2015)) extends this idea to mobile devices by combining the orientation tracking and touch-sensitive 2D surface of a handheld device. The orientation of the device defines a virtual plane centered at the object's location, while translation of the user's fingers on the 2D surface is interpreted as displacement of the virtual object on the predefined plane. INSPECT facilitates different control modes where the pivot mode fixes the plane at the original position of the object thus any further rotation results in a displacement of the object, while the free-plane casting mode always places the pivot of the plane at the location of the virtual ob-

ject, thus any displacement must be initiated through swipe gestures on the handheld device. Piekarski and Thomas (2004) also introduced a plane-based system on mobile devices, enabling 3D modeling and manipulation of distant AR content.

Though existing methods enable high-DoF positioning in the virtual environment, most of them require extensive hardware, e.g. mice or touch panels, which are difficult to apply to hands-free interactions.

3.2.2 Gaze-supported Interaction

When users are interested in an object, they tend to look at that object. This tendency suggests that eye gaze can be a natural modality for interaction with virtual content. It has commonly been used for selection in 2D and 3D environments (Johansson et al. (2001); Stellmach and Dachselt (2013)). Advancement in eye tracking technology has led to a series of studies that compare the performance of these targeting techniques with each other in terms of accuracy and speed. Kytö et al. (2018) showed that using only eye-gaze to target and select targets generally performs slower and less accurately than head gaze or a combination of eye gaze with other modalities. These results confirm previous findings by Qian and Teather (2017), who found that head-gaze was more reliable than eye-gaze. At the same time, Blattgerste et al. (2018) found that eye gaze outperformed head gaze in terms of accuracy, speed, and task load. They also found that this advantage was more dominant in HMDs with a larger field of view. They attributed this conclusion to more reliable eye tracking in their evaluation. It is thus still unclear which method will perform better in the long run.

After a target is selected with either eye gaze or head gaze, other methods allow for manipulation of the object through hand gestures (Stellmach and Dachselt (2013); Pfeuffer et al. (2017); Song et al. (2014)) or other controllers. While few 2D interfaces use eye gaze for selection and manipulation (Hornof et al. (2004)), no technique appears to be available that allows object manipulation in 3D using only eye gaze. This can be traced back to a variety of reasons. Compared to interaction with 2D interfaces, AR and VR present additional challenges for object selection and manipulation, for example handling occluded objects or those in the same line of sight. The focus depth can be derived either from the vergence of the user's gaze (Lee et al. (2017)), by selecting the object of interest from a list (Piumsomboon et al. (2017)), or combining gaze with other modalities (Mardanbegi et al. (2019)). Although the above methods can disambiguate between objects at different depths, the

estimation is too coarse to accurately manipulate the object depth. Furthermore, it is difficult to manipulate the depth of an object after it was selected.

3.2.3 Further Motivation

Unlike other studies that use gaze either as a confirmation method or to support hand interactions, this work is dedicated to provide an tool that could be carried out in an entirely hands-free manner. The goal for this work is to test the designed interface to see how well it could help users intuitively manipulate a virtual object. Though OrthoGaze can also potentially be adapted for rotation and scaling, this work thoroughly examines the fundamental properties of the method and its usability for translation tasks. Furthermore, since previous studies found contradicting results on the efficiency of eye and head gaze while focusing on 2D selection scenarios such as menus, it is also necessary to study how these techniques perform in 3D scenarios where users have to interact with objects at different depths.

3.3 Methodology

This section firstly discusses the process behind the development of a gaze-only manipulation method. Because of constraints of the human visual system, the design of OrthoGaze is fundamentally different from typical 2D or touch-interface methods. This section then describes the design and use of OrthoGaze and why this works well for manipulating virtual objects. In general, all units discussed in the following sections are relative to the world coordinate system.

3.3.1 Constraints for Gaze-based Object Manipulation

When testing and brainstorming different methods for gaze-based object manipulation, two primary constraints were identified that needed to be accounted for. The first is related to one of the inherent characteristics of gaze fixations. Johansson et al. (2001) showed that gaze plays an important role in leading movement during manipulation tasks. While a user is performing a sustained manipulation of an object, as a natural response, his/her gaze will tend to be fixated on the object as it travels to ensure that the manipulation is carried out correctly. Thus, an interface should cause the smallest possible eye gaze offset from the target object during the manipulation process. In other words, the method benefits greatly from synchronously carrying out the

manipulation with the user's gaze, rather than with a clutch or secondary mechanism.

Another main constraint is that although it is possible to estimate the approximate gaze depth through eye tracking (Lee et al. (2017); Mardanbegi et al. (2019)), consciously and precisely adjusting one's gaze to an arbitrary depth without guidance is very difficult. This highlights the necessity of providing a clear method to guide the user's eyes, especially to a particular depth in 3D space.

3.3.2 Orthogonal Plane Design

OrthoGaze is composed of three orthogonally intersecting square planes, as shown in Fig. 3.2. Each plane indicates a 2-DoF space for manipulation in virtual world coordinate space. For instance, the yellow plane in Fig. 3.2 denotes the world xy plane. This is a per-object design, where the three planes are attached to each object at its geometric center.

The planes fulfill two roles during the manipulation. First, as noted in Section 3.3.1, it is hard for an individual to focus his or her gaze at a particular point in 3D space without a salient feature or object to focus on. As such, the planes provide a surface onto which the user can focus to adjust the position of an object. Second, the plane constrains the object movement to within the plane even if other objects are in the user's view. The three orthogonally intersecting planes allow users to manipulate all 3 positional DoF where each DoF can be manipulated by two planes. This design also ensures that users always have access to at least one plane even at a poor viewing angle, mainly when it is perpendicular to the user's gaze. Additionally, to maintain general visibility and accessibility, the size of all planes s_{pln} is scaled linearly based on the distance to the object d_{obj} as:

$$s_{\text{pln}} = s_{\text{bsc}} \times (1 + p \times d_{\text{obj}}) \tag{3.1}$$

where s_{bsc} denotes the size when $d_{\text{obj}} = 0$ and p is a constant scaling factor.

3.3.3 Manipulation Mechanisms

OrthoGaze assists object manipulation through two main functions:

Plane activation In the default state all planes appear semi-transparent, indicating that the user can trigger an interaction by looking at them. When the gaze ray intersects with a plane, that plane is highlighted to indicate to the user the detected selection. If the user's gaze remains on the plane throughout

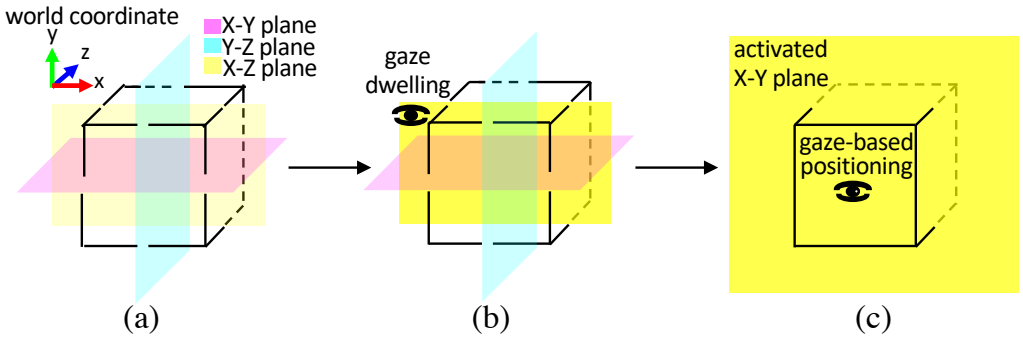


Figure 3.2: Images showing an example of the baseline design of OrthoGaze. (a) Three orthogonal planes with different colors intersecting at the geometric center of the object, waiting for activation. (b) Highlighted X-Y plane by user's gaze dwelling. (c) Activated X-Y plane where user can position the object at a 2-DoF level.

the dwell period, the system switches into object manipulation on this plane. This trigger is based on whether or not the gaze ray is intersecting with a given plane. Therefore, as long as the gaze point stays on that plane, the user can still activate the manipulation even if the gaze is jittering or if the estimation is slightly inaccurate.

2-DoF Manipulation Once a plane is activated, its size expands to better represent the positionable area, and the remaining planes temporarily become transparent to allow seamless manipulation with the selected plane. The object will then follow the intersection point of the user's gaze ray and the plane. This ensures that the object will always be at the location the gaze is focused on.

Placement Finally, placement at the destination is triggered by another gaze dwell. After this placement is completed, the system switches back into its default state that shows all planes as semi-transparent.

To trigger the activation and selection, gaze dwell is detected as follows. In eye gaze mode, gaze dwell at time t_0 is calculated as an angular deviation of eye gaze OE over a time period n:

$$OE_{t_0} = \frac{1}{n} \sum_{t=t_0-\mathrm{n}}^{t_0} \| \arccos(\hat{eye}_t \cdot \hat{head}_t) - \arccos(\hat{eye}_{t-1} \cdot \hat{head}_{t-1}) \| \quad (3.2)$$

where $\hat{eye}$ is a unit vector of eye gaze, $\hat{head}$ is a unit vector of head gaze and $\cdot$ is an operator of the inner product between vectors. The eye gaze is assumed to be fixated at a location if OE is less than a threshold ts. Through initial

tests, it is found to be more robust to detect natural eye gaze dwell using such angular deviation rather than the exact gaze point.

In the head gaze mode, a dwell is detected if the head gaze deviation $OH \leq$ ts, where

$$OH_{t_0} = \frac{1}{n} \sum_{t=t_0-\mathrm{n}}^{t_0} \| \arccos(\hat{head}_t \cdot \hat{head}_{t-1}) \|. \tag{3.3}$$

Though a number of different selection methods are compatible with OrthoGaze, gaze dwell was chosen to avoid the Midas Touch problem (Jacob (1995)) because it is easy to understand and prevalent in research. This decision was also made as a trade off with accuracy to reduce necessary eye movements and maintain intuitiveness.

3.4 User Study

The user study was aimed to investigate the efficiency and the effectiveness of OrthoGaze. In addition, it is also necessary to test how head, eye, and controller based manipulations would perform using this interaction paradigm. To test this, two different tasks, painting (A) and docking (B) for simplicity, were implemented to evaluate OrthoGaze both fundamentally and practically. When participants entered the experiment room they first received an introduction into the experiment tasks and an explanation of the different control modes of OrthoGaze. After signing a consent they first completed task A followed by task B. During the experiment participants remained seated on a swivel chair and were asked not to stand up or move around. However, local body movements were not physically restricted, and participants could rotate their chair if necessary, which is natural when performing interactions in VR. Participants could take a break between each trial if needed. Participants had a training session before each task where they could practice all of the designated methods. After all trials, participants completed a custom survey (A.1) related to the task and their experiences. Overall, the experiment took about 1 hour. The procedure of the experiment was approved by the institutional review board of Osaka University Review Board.

3.4.1 Hardware and Participants

For the evaluation, an HTC Vive Pro Eye was used as the HMD, which has integrated eye tracking cameras and provides relatively stable eye gaze data. A virtual environment with the experiment tasks was set up using Unity

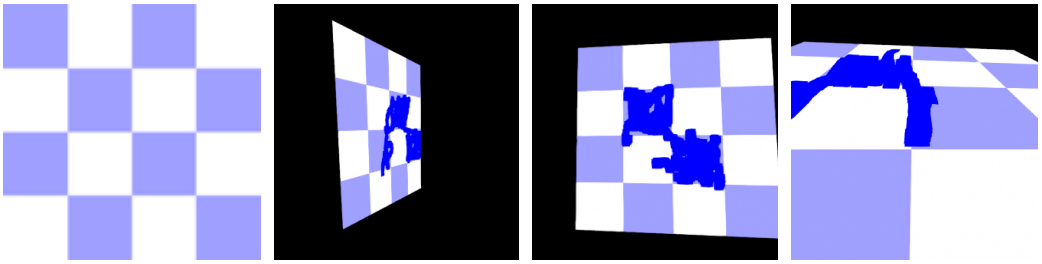


Figure 3.3: Images showing the content of task A. (a) The chessboard plane used in the experiment. Participants were asked to paint the purple blocks as fully as possible and avoid spilling onto the white blocks. Sample frames of a participants' views during the task are shown and represent the (b) Y-Z plane, (c) X-Y plane, and (d) X-Z plane.

2018.3.2f1. The whole system was run on a desktop computer with an Intel Xeon E5-2690 CPU and an NVIDIA GeForce GTX 970 GPU, at an average frame rate of 60 frames per second. R 3.6.2 [1] and coin 1.3-1 [2] were used for the statistical evaluation.

18 students and researchers were recruited from the local university, 15 male and 3 female, ranging in age from 21 to 38 (avg. 25.56, stdev. 4.22). 9 of them wore prescription glasses during the experiment. 10 of them had no experience in eye-based human-computer interaction, while the remaining 8 had some experience (less than 5 times in total) before this experiment. All participants received a gift card worth approximately 5 USD as remuneration.

For all eye gaze methods, the participant's neck was not restricted, which means the participants were allowed to perform natural eye movements supported by head movements.

3.4.2 Task A: Gaze-based Painting

As described in Sec. 3.3, OrthoGaze presents multiple planes in the world coordinate system and allows users to translate objects within each plane for a continuous period of time. In other words, the accuracy and speed with which users can sustain gaze action can significantly affect performance. As such a task was designed to compare aiming with head versus eye gaze.

[1] https://www.r-project.org/
[2] http://coin.r-forge.r-project.org/

3.4.2.1 Hypotheses

Some evidence supports the hypothesis that head gaze has better performance than eye gaze for discrete selection tasks in VR environments (Qian and Teather (2017); Kytö et al. (2018)). In contrast to discrete tasks, OrthoGaze requires continuous gaze movements in a large, dynamic range of distances, i.e. 3D manipulations. Therefore task A was aimed to compare eye and head gaze under continuous targeting. Following the insights from previous work that examined gaze performance for discrete tasks and the expected higher stability of head gaze compared to eye gaze, it was hypothesized that:

Ha1 Head gaze will allow for faster performance than eye gaze for distant, continuous aiming tasks.

Ha2 Head gaze will be more accurate than eye gaze for distant, continuous aiming tasks.

3.4.2.2 Task

The experiment was conducted in a virtual environment with a solid black background. To evaluate the accuracy and the speed of sustained eye gaze and head gaze behaviours, participants had to aim at indicated areas of a target plane while avoiding other areas of the target. A 4×4 chessboard pattern with 2 colors was used as as the target plane as shown in Fig. 3.3(a). The checkerboard appeared at 3 different positions in the world coordinate system for each gaze method, and each position represented a different 2-DoF plane. Detailed specifications of the environmental setup are shown in Table. 3.1. Note that the setup of the planes was selected empirically to cover the specific fields of view of the HMD. This avoided severe view point shifts caused by view point changes. For instance if the left plane were placed on the participant's left flank, it might appear the same as the front plane if he or she turned left.

For a single trial, participants were required to use their head gaze or eye gaze to aim at and cover as much as possible of the purple region on the plane

Table 3.1: Environment specifications of task A.

brush size	plane size	position (x, y, z)	rotation (x, y, z)
4×4	120×120	left $(-100, 1, 170)$ front $(0, 1, 170)$ ground $(0, -21, 60)$	left $(-90, 0, 0)$ front $(0, 0, -90)$ ground $(0, 0, 0)$

* Each plane is pivoted on its geometric center.
* Participants' viewing point (head position) is located at $(0, 1, 0)$.

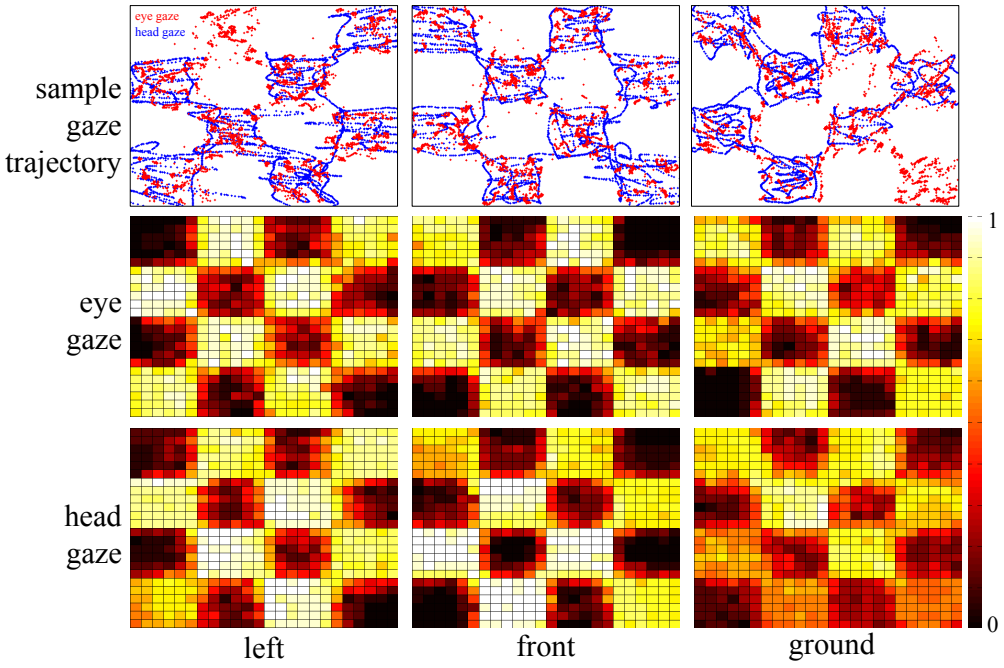


Figure 3.4: (top) Sample gaze points from a participant, projected onto 2D planes. (middle and bottom) Images showing heat maps of gaze point coverage results of task A over all participants. Note: to represent frequency of gaze among all participants, each segment containing gaze points is counted only once for each participant regardless of the total gaze points for that participant. The brighter a region, the higher the gaze frequency.

within 60 [s], while avoiding aiming at the white region. The intersection point of the gaze and the plane functioned as a square painting brush in blue. Participants could not deactivate the painting during the trial, which means participants would be painting whenever their eye gaze or head gaze intersects with the plane. The experiment was conducted as a 2×3 within-subject study with 2 gaze methods (eye and head gaze) and 3 plane orientations (left, front, and ground) which resulted in 6 trials for each participant. The order of the trials between participants was randomized. Before each trial participants saw an outline of the plane location in the next trial and were informed which gaze technique will be used. The experimenter toggled the next trial at which point the plane became visible and users could paint the plane.

3.4.2.3 Results

Participants' eye gaze point as well as the head gaze point were collected for each frame when the gaze intersected with the target plane. From the collected

data, trajectories were computed consisting of both eye gaze and head gaze points, as shown in the top row of Fig. 3.4. For quantitative evaluations, each plane was segmented into 24×24 regions, and evaluated the speed and the accuracy of each gaze method by calculating the cover rate and the ratio of gaze points falling onto target areas versus non-target areas. An Anderson-Darling test showed that the data was not normally distributed, thus the Friedman's test was used to analyze variance when necessary, and the Wilcoxon signed rank test was used for post-hoc tests. A threshold of $p = 0.05$ was used to determine statistical significance. $r = Z/\sqrt{N}$ is reported as the effect size for all (post-hoc) test results, where Z is the statistical value and N is the total sample size (Rosenthal (1994)).

For each participant, the cover rate is given by the ratio of covered correct area compared to the overall target area. In other words, this represents how much of the purple area participants painted blue. As all participants had the same amount of time to cover as much of the target area as possible, the cover ratio represents the speed with which participants can aim and adjust their gaze in a continuous task.

As described above, the plane was segmented into 24×24 regions. If at least one gaze point fell into a region, it would be counted as covered for this participant. The middle and bottom rows in Fig. 3.4 show results of cover rates of each plane condition for all participants visualized as frequency heat maps. More detailed results of the gaze cover rate are shown in Fig. 3.5(a). Although no statistically significant difference was found for cover rate when participants aimed on left ($Z = 0.762, r = 0.13, p > 0.05$) and front plane ($Z = 0.305, r = 0.05, p > 0.05$), participants covered significantly less area with head gaze than eye gaze for the ground plane ($Z = 2.308, r = 0.38, p < 0.05$). In general, no significant difference was found when comparing the cover rate of both gaze methods, regardless of the aimed plane ($Z = 1.270, r = 0.12, p > 0.05$).

The accuracy of the gazing was defined as the ratio of gaze samples that fall into the correct areas over all gaze samples for a participant (Fig. 3.5(b)). There were significant differences between eye gaze and head gaze for the left ($Z = 2.722, r = 0.45, p < 0.01$), front ($Z = 2.896, r = 0.48, p < 0.01$) and ground ($Z = 3.070, r = 0.51, p < 0.01$) plane. In all cases participants were more accurate when using eye gaze than head gaze, which is also supported by the data aggregated by method ($Z = 4.886, r = 0.47, p < 0.001$).

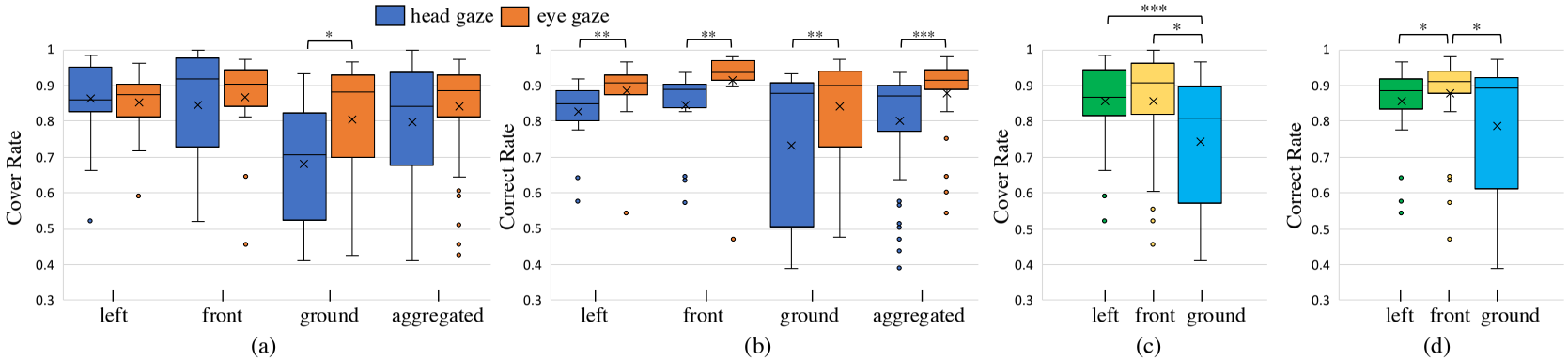


Figure 3.5: Boxplots showing the quantitative results of task A. (a) The segmental cover rate of each plane, along with data aggregated by method. This is calculated based on a 24×24 segmentation of the target planes. (b) The correct rate for aiming on each plane, along with data aggregated by method. This is the rate of gaze points that fell into correct segments out of all collected points. (c) The cover rate aggregated by plane. (d) The correct rate aggregated by plane. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

For the results regardless of the gaze method (Fig. 3.5(c) and (d)), a Friedman's test showed a significant difference between each plane both in the cover rate ($\chi^2(2) = 20.72, p < 0.001$) and the correct rate ($\chi^2(2) = 8.39, p < 0.05$). For the cover rate, a post-hoc Wilcoxon signed rank test with Bonferroni correction showed a significant difference between left and ground plane ($Z = 3.456, r = 0.41, p < 0.001$), and between front and ground plane ($Z = 4.313, r = 0.51, p < 0.001$). For the correct rate, significant difference was found between left and front plane ($Z = 2.671, r = 0.31, p < 0.05$), and between front and ground plane ($Z = 2.765, r = 0.33, p < 0.05$).

Participants were also asked to rate helpfulness, ease of use, and fatigue for each method, but no statistically significant results was found for each item.

3.4.2.4 Section Discussion

The results reject hypotheses **Ha1** and **Ha2**. This result is different from previous findings that showed that head gaze outperforms eye gaze for selecting near-field targets (Qian and Teather (2017)). One potential reason for this is that the longer distance between the target and the user could have led to higher difficulty of head gaze aiming compared to eye gaze, as users had to utilize more muscles. When aiming on targets that are far away, finer control is needed for head gaze since even small angular movement of the head can cause a huge offset projected in the distance, while eye gaze can remain robust in aiming accuracy since users aim with their eye gaze by directly looking at the target position. Additionally, aiming was significantly less accurate on the left and ground planes, as indicated by Fig. 3.5(d), which could have resulted from the reduced perspective since the front plane was parallel to participants' view and thus had better visibility compared to the other two. It was also observed that participants had more difficulty aiming on the ground plane than the other planes, as highlighted by the middle and lower right of Fig. 3.4 and Fig. 3.5(c) and (d). It is believed that this is because it was hard to rotate the neck to cover all areas of the plane, as the ground plane was set to occupy the area right at the participant's feet.

3.4.3 Task B: Three-dimensional Docking

While task A evaluates the very basic performance of gaze-based methods interacting with orthogonal planes, task B evaluates the actual usability of OrthoGaze for manipulating the 3D position of virtual objects. As OrthoGaze is intended for gaze-based interaction, it generally has good compatibility for

raycast-based interaction. Thus task B was designed to compare the performance of OrthoGaze with eye gaze, head gaze and a controller. While dwell timing was used to trigger the different modes for eye gaze and head gaze, raycast aiming coupled with button clicks was used in the controller condition. In that sense, the controller condition serves as a best-case benchmark.

3.4.3.1 Hypotheses

From observations in Task A, it was expected that participants could utilize OrthoGaze better with eye gaze than head gaze as it was more accurate and faster. Furthermore, as the controller condition did not suffer from the constraint of the dwell time and possible unintentional activation, it was expected to outperform other control conditions. Overall the following hypotheses were set for task B:

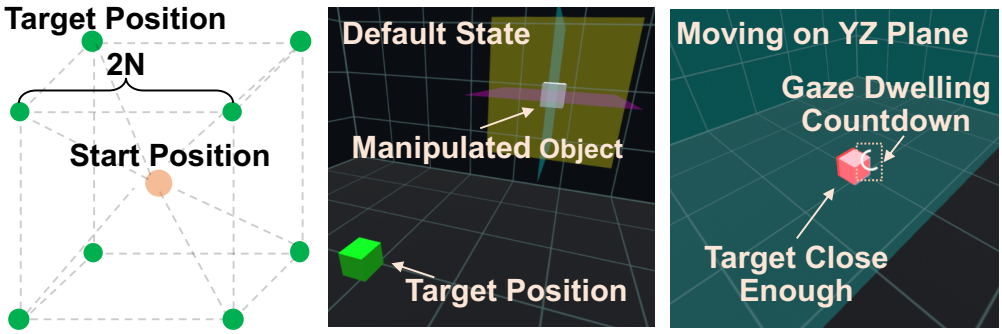
Hb1 Participants will successfully complete the docking task more often and more quickly when using the hand-held controller than head and eye gaze.

Hb2 Participants will perform the docking task faster with eye gaze than head gaze.

3.4.3.2 Task

For task B, participants were located at $(0[\text{m}], 1[\text{m}], 0[\text{m}])$ and had to move a white cube, sized $0.5[\text{m}] \times 0.5[\text{m}] \times 0.5[\text{m}]$, from a fixed start position $(-1[\text{m}], 0.5[\text{m}], 5.5[\text{m}])$ to several target positions using all three control methods described above with OrthoGaze. During each trial, a green cube with the same size as the white cube appeared at one of the target locations and participants had to align the white and the green cubes (Fig. 3.6(b)). The target positions were corners of an imaginary cube with a side size of $2N[\text{m}]$ whose center coincided with the center of the white cube as shown in Fig. 3.6(a). 8 imaginary cube sizes with $N \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$ were used. To keep the number of alignments reasonable each offset direction was selected twice and was paired with a different distance in each appearance thus ensuring that each offset distance and direction appeared twice during the experiment, resulting in overall 16 different target positions.

Since it is difficult to perfectly align the white and green cubes, a successful alignment is confirmed if the two cubes are less than 0.2m apart when the user confirms the placement. As an additional cue the target cube will turn from semitransparent green to red when the two cubes are within the threshold



(a) (b) (c)

Figure 3.6: Images showing the content of the docking task. (a) 8 target positions distributed from the start position equally in world x, y and z axes. Note that 16 target positions were used in total, paired with 8 different distances from the start position. (b) A sample of a start condition of task B. (c) Moving the cube close enough causes a change in the target cube from green to red. Not only moving the cube to the target position but also placing it successfully within 30 [s] counts as a successful trial.

distance of each other (Fig. 3.6(c)). Note that to succeed in a trial, the participant not only had to move the cube to the target position, but also had to perform the placement successfully. If the trial was a success, the participant received a sound effect as confirmation.

Before starting the experiment, participants had 10 trials for practicing with each control method. During the experiment, participants initially saw an empty room. Each trial started when the white cube and the target location appeared in front of the participant and finished after successful alignment. If participants did not align the cubes within 30 [s] the trial was counted as failed.

For utilizing OrthoGaze with gaze-based methods, the dwell time n was set to 1.3 [s] and the constant threshold ts for the angular gaze offset to 0.005 [rads] for both plane activation and object placement. When using the controller, participants used a raycast to aim on planes and pressed the trigger button for activation and placement. This sets the controller as the standard of utilizing OrthoGaze with least time loss, to which the performance of gaze-based methods could also be compared. This experiment was conducted as a 3×1 within-subjects experiment with the control method as independent variable. The order of the conditions was counter-balanced for all participants using a Latin square.

For evaluating the experimental results quantitatively, the following metrics were recorded and calculated:

- **Success rate**: The success rate is calculated for each participant as the rate of successful trials out of all trials. This evaluates the general efficiency of manipulating objects with OrthoGaze, since both accuracy and speed are comprehensively required to successfully complete a trial.
- **Completion time**: Completion time is recorded for each successful trial, and is ignored for failed trials. Note that for gaze-based methods, the completion time was recorded with the time of performing gaze dwell both included and excluded.
- **Final distance**: Final distance is recorded only if the participant fails a trial, normalized as the final distance divided by the initial distance. This normalization stands for how close/far the participant managed to move the object to the target with regards to its initial position.

3.4.3.3 Results

For the success rate (Fig. 3.7(a)), both the head gaze mode (avg. 0.934, stdev. 0.067) and the controller mode (avg. 0.969, stdev. 0.052) reached over 0.9 average success rate, while the eye gaze mode (avg. 0.757, stdev. 0.224) reached over 0.7 with a relatively high deviation. An Anderson-Darling test showed that the success rate was not normally distributed, thus it was evaluated with a Friedman's test that showed statistical significance between the different modes ($\chi^2(2) = 21.73, p < 0.001$). A post-hoc Wilcoxon signed rank test with Bonferroni correction showed significant differences between head gaze and eye gaze mode ($Z = 3.271, r = 0.55, p < 0.001$), and between eye gaze and controller mode ($Z = 3.424, r = 0.57, p < 0.001$). This result shows that the controller significantly outperforms eye gaze, which partially supports hypothesis **Hb1**, and head gaze also significantly outperforms eye gaze, which rejects hypothesis **Hb2**.

For the completion time including dwell time (Fig. 3.7(b)), eye gaze mode took the longest (avg. 16.368 [s], stdev. 5.816) over head gaze mode (avg. 13.351 [s], stdev. 5.097) and controller mode (avg. 9.526 [s], stdev. 4.829). Anderson-Darling test showed that the success time was not normally distributed. Since the data include different sample sizes, a Kruskal-Wallis test was applied and showed statistical significance between the different modes ($\chi^2(2) = 193.47, p < 0.001$).

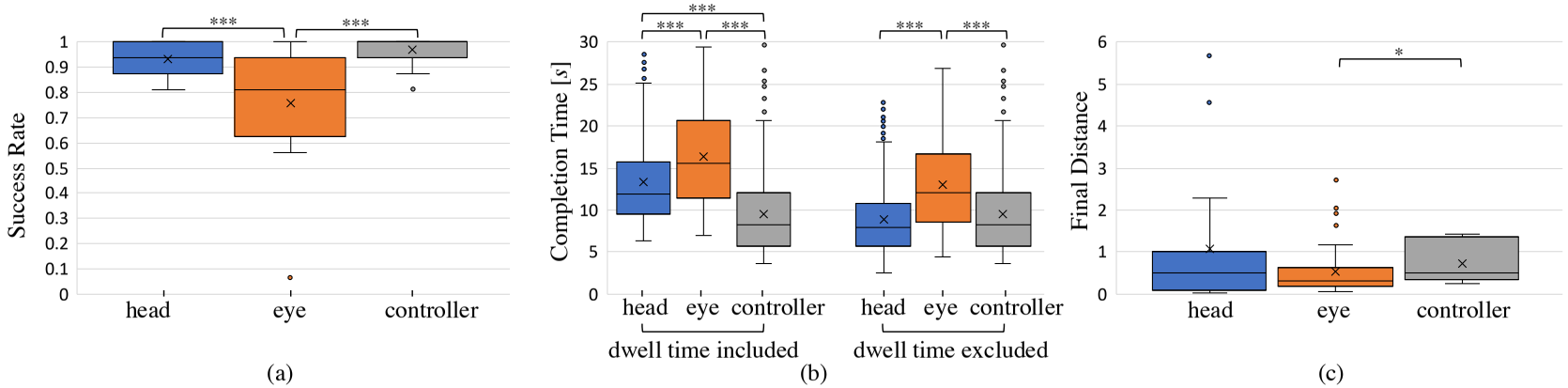


Figure 3.7: Box plots showing the quantitative results of Task B. (a) Success rate of the docking task (1 = 100% success) for each control method. (b) Completion time of successful trials for each control method, with the dwell time included (left) and excluded (right) for gaze-based methods. (c) Final distance between the moving cube and the target of failed trials, where each distance is normalized based on the initial distance of that trial. (***: $p < 0.001$, *: $p < 0.05$)

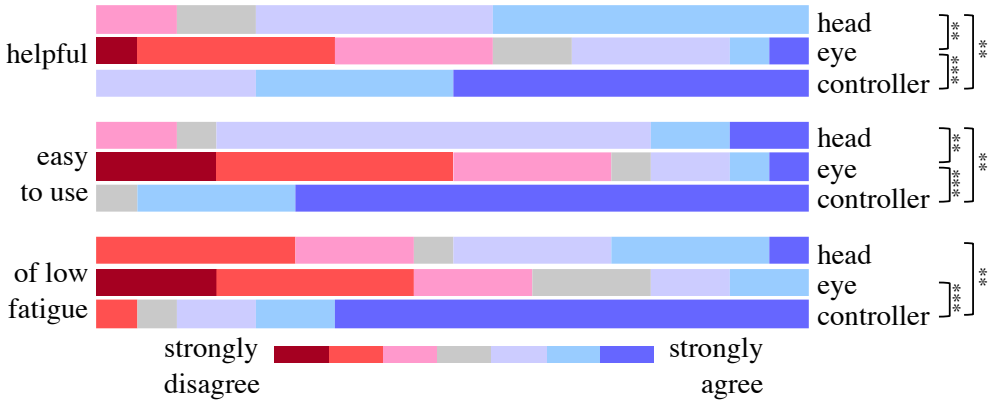


Figure 3.8: A 7-point Likert scale chart showing the subjective results regarding the user experience of each control method in task B. (***: $p < 0.001$, **: $p < 0.01$)

A post-hoc Mann-Whitney U test with Bonferroni correction showed significant differences between head gaze and eye gaze mode ($Z = 6.065, r = 0.27, p < 0.001$), head gaze and controller mode ($Z = 9.678, r = 0.41, p < 0.001$), and eye gaze and controller mode ($Z = 12.689, r = 0.60, p < 0.001$). The reciprocal of completion time was also computed as an evaluation of completion speed. On average, head gaze achieved a 71% and eye gaze achieved 58% performance of completion speed in comparison to the controller mode. Interestingly, with the dwell time excluded ($\chi^2(2) = 91.91, p < 0.001$), significant difference was only found between head gaze and eye gaze mode ($Z = 8.929, r = 0.40, p < 0.001$), and eye gaze and controller mode ($Z = 7.855, r = 0.35, p < 0.001$). In this case, head gaze achieved a 107% and eye gaze achieved 74% performance of completion speed compared to the controller.

Fig. 3.7(c) shows the final distance between the moved position and the target object of the last frame for all failed trials of each method. A Kruskal-Wallis test showed statistical significance between the different modes ($\chi^2(2) = 11.37, p < 0.01$). A post-hoc Mann-Whitney U test with Bonferroni correction showed significant differences only between the eye gaze and the controller modes ($Z = 2.816, r = 0.23, p < 0.05$).

The results of the subjective questionnaire were shown in Fig. 3.8. A Friedman's test revealed significant differences in helpfulness ($\chi^2(2) = 21.34, p < 0.001$), easiness ($\chi^2(2) = 20.38, p < 0.001$), and fatigue ($\chi^2(2) = 22.53, p < 0.001$). A post-hoc Wilcoxon signed rank test with Bonferroni correction showed that for helpfulness significant differences were found between head

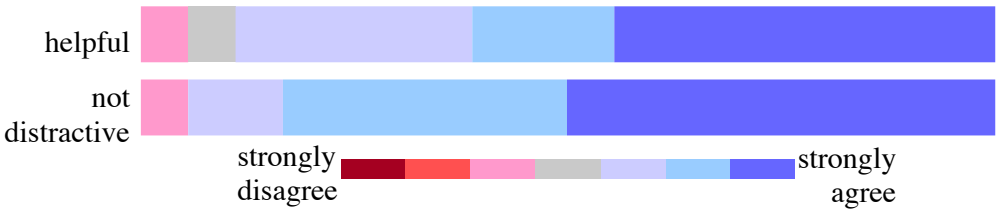


Figure 3.9: A 7-point Likert scale chart of the subjective evaluation regarding the usability of OrthoGaze.

gaze and eye gaze ($Z = 2.887, r = 0.48, p < 0.01$), eye gaze and controller ($Z = 3.517, r = 0.59, p < 0.001$), and head gaze and controller modes ($Z = 2.962, r = 0.49, p < 0.01$). For the easiness, significant differences were found between head gaze and eye gaze ($Z = 3.315, r = 0.55, p < 0.01$), eye gaze and controller ($Z = 3.521, r = 0.59, p < 0.001$), and head gaze and controller modes ($Z = 3.358, r = 0.56, p < 0.01$). In regards to fatigue, significant differences were found between eye gaze and controller ($Z = 3.422, r = 0.57, p < 0.001$), and head gaze and controller modes ($Z = 3.316, r = 0.55, p < 0.001$).

As a general result for each item of the survey, the controller mode was significantly considered of highest subjective scores, which also supports hypothesis **Hb1**, followed by the head gaze mode, while the eye gaze mode had the lowest.

Lastly, participants were asked to answer two questions about their experience with OrthoGaze, in particular if it was helpful and if it's visually distracting, along with freeform feedback. Fig. 3.9 shows the answers. 16 out of 18 participants thought that OrthoGaze helped them complete the task and 17 participants did not think that OrthoGaze was distracting.

3.4.3.4 Section Discussion

The results of task B support hypothesis **Hb1**. It is believed that this could be in part due to the different confirmation mode utilized in controller mode as participants noted that it was more difficult to keep their gaze fixated on the target for a long time. Note that the threshold selected to confirm a selection was a very conservative value since it was intended to avoid unintentional triggering and a shorter threshold could make dwell timing more comfortable. The overall positive rating of OrthoGaze being helpful suggests that the improvement in its confirmation mode could potentially further improve the performance with both gaze-based methods.

Interestingly, the results rejected hypothesis **Hb2**. From the participants'

comments and the feedback of the questionnaire, it is believed that the main limitation in confirming a selection was the gaze dwell. Another explanation could be that each task was rather quick and did not require continuous movement within the plane, but short movements to a new location before confirming the selection. From Fig. 3.7(c), for the failed trials, eye gaze-based control had the closest average distance from target. This result reflects the situations in which the participant could correctly move the cube to the target position but could not manage to place it, which was also reported as oral feedback. This supports some previous results that eye gaze is less stable for short-term actions such as selections compared to other methods (Qian and Teather (2017)). One conceivable approach to solve this issue is to use a combination of eye and head gaze, e.g. using eye gaze assisted by head gaze for moving and pure head gaze dwelling for placement.

The subjective results shown in Fig. 3.8 suggest that using gaze-based control for accomplishing manipulation tasks over a sustained period of time can cause higher fatigue than controller-based control. Subjective feedback indicated that this higher fatigue of using gaze for long-term tasks likely comes from the requirement for higher concentration when consciously controlling the gaze. In addition, there were verbal comments regarding usability such as "*It was difficult to perform the eye gaze dwell*", "*The eye gaze was difficult to fixate, but it's convenient for moving the object*", "*The gaze dwell time was long/short*", and "*The unconscious attempt for the next action sometimes caused an eye movement and thus reset the gaze dwell timer*". It is necessary to further investigate the tasks for which head and eye gaze perform best and how to best combine these gaze techniques to achieve better performance.

3.5 Discussion

This section discusses the findings, challenges and limitations regarding OrthoGaze and the experimental designs, conceivable future improvements, and directions in detail for further research.

3.5.1 Implication

The results of task A suggest that although participants can cover areas far away from the user using head gaze, it comes at the cost of lower accuracy as the gaze is more susceptible to minute head movements. At the same time targeting areas at the user's feet could result in lower speed due to a restriction of the neck's movement range. Similarly to head gaze, a decrease was observed

in the performance of the eye gaze on the ground plane compared to the other planes, although this decrease was less prominent than for the head gaze. For the ground plane the larger movement range of the eyes in combination with head movement allowed users to cover more of the target plane. These results suggest that it is necessary to pay careful attention to the area users will interact in with the virtual content and the required accuracy.

For task B, it was observed that some participants could move the cube to the target position, but still failed the trial because they had difficulty gaze dwelling for the final placement. There were also some oral comments from the participants saying that the dwell time was too long or short, which indicates the existence of individual differences on the preference of gaze dwell time that can potentially affect the user experience. The placement method is essential to the performance of manipulation tasks, and the use of gaze dwell likely reduced overall performance and subjective user experience such as fatigue. The completion time excluding gaze dwell time also indicates that the performance of OrthoGaze may have appeared sub-optimal in experiment results and could likely benefit from fine tuning with the method of selection. As a next step, one would test the usability and the efficiency of different windows of gaze dwell time for both activation and placement through user study for developing a method that can achieve more stable placement through eye gaze and suit individual preference for gaze dwell time. It is also valuable to explore other selection methods that can specifically improve the usability of manipulation using eye gaze.

The results of task B showed that OrthoGaze was able to facilitate eye and head gaze-based manipulations with 78% and 96% success rates in comparison to a hand-held controller, respectively, though there still existed latent improvement in its baseline design. As participants of the experiment were relatively unfamiliar with eye gaze-based user interface, it is also expected that OrthoGaze has the potential to perform better if users are more practiced.

In short, from the results of the user study, OrthoGaze is expected to become an effective method to handle pure hands-free object manipulations. For individuals with handicaps or for users whose hands are constantly occupied with other work, OrthoGaze can provide an effective way to conduct 3D manipulations moving forward.

3.5.2 Limitations

One critical factor that can affect the performance of OrthoGaze is the viewing angle of the planes. In the docking task, participants were asked to stay seated

to exclude the effects of large body movements. However, this constraint sometimes reduced the performance of the gaze-based methods compared to the controller in an unfair way. In situations where the participants had small viewing angles, a few participants managed to finish such trials skillfully by first adjusting on another plane to acquire larger viewing angle, while most participants attempted to move their body to physically change the viewpoint but still failed the trails since they had to stay seated. However, they could still manage the aiming using the controller by reaching out their arms to extend the incident angle of the raycasting. Thus it is hypothesized that OrthoGaze can achieve higher usability if the user is able to change his/her viewpoint, or if the viewing angles of the planes are adjustable based on the user perspective, which needs to be validated through further user study.

In addition, in the docking task OrthoGaze was simply visualized as a set of semi-transparent planes. This was set in order to test OrthoGaze's naive design. However, the different texturing of the planes in the two tasks could also have affected performance. For example, the grid-textured planes could potentially help with guiding the user's eyes via certain distance cues and hence improve the aiming accuracy of eye gaze. It was also observed that in some cases participants had difficulty in activating the plane. This was due to the fact that the complete visualization of all planes might occlude each other and thus prevent the user from properly interacting with the planes. This would also make it difficult to start the manipulation at the original position, and thus make it hard to perform slight refinements of the position. Such limitations can also be prevented by optimizing the visualization, for example by adaptively visualizing the optimal interactive area based on the user's view point or projecting a shadow copy of the moved object for reference. Moreover, a two-step confirmation could also help address this issue, for example defining an area on the plane further away from the object in which the user needs to focus his or her gaze and then adjusting the gaze back to the object for completing an activation. As a next step, it is planned to refine these designs and test the effects of different plane visualizations on performance.

The design of the user study for OrthoGaze also has several potential improvements. Since task B was mainly intended for testing the usability of OrthoGaze, the evaluation was emphasized in a task-based manner, i.e., the success rate was evaluted and a threshold of within 0.2 [m] was set as successful docking. Though the normalized final distance was also evaluated, this evaluation is difficult to make completely fair as it only counts for failed trials and it is based on the assumption that the difficulty of all trials are equivalent. Therefore, the study lacks evaluation of precise accuracy. In addition,

the docking task was conducted in a widely open VR environment where little visual occlusions could occur. Considering the visualization of OrthoGaze, the complexity of the environment also could impact the performance, as the planes of OrthoGaze could occlude other objects in the environment and thus affect the user experience. A complex background could affect the visibility of the planes as well. In short, further study needs to be done regarding the accuracy control of OrthoGaze, as well as visual effects in more complex and practical environments.

3.5.3 Future Work

OrthoGaze can be potentially extended to hands-free manipulation of even higher DoF, including but not limited to continuous rotation and scaling, which will be meaningful to achieving hands-free object modeling for large VR and AR environments. However, such functions could also be accompanied with the issue of lost gaze focus. As stated in Section 3.3.1, if the feedback of the manipulation is not appropriately synchronized with gaze movement, it may result in confusion with the correct recognition of the manipulation result. This is a big challenge for the visualization, proper function, and intuitiveness of manipulations such as rotations.

Additionally, OrthoGaze can also be applied to optical see-through HMDs to achieve hands-free manipulation in AR. In some outdoor AR use cases, such as on a crowded train or in a theatre, it is usually not preferred to use gestures or voices as interaction tools, which meets the requirement for hands-free interaction. To apply OrthoGaze to optical see-through AR, visualization of the planes will likely need to be redesigned in comparison with the VR use cases since it is important to preserve the visibility of the real world.

3.6 Chapter Conclusion

This work introduces OrthoGaze, a novel approach that allows users to manipulate the 3D position of virtual objects in a virtual environment using eye or head gaze alone. The method is composed of three orthogonal planes, which are affixed to the geometric center of a target object during manipulation. Users can activate each plane using their eye or head gaze, and then move the object on the activated plane by matching the intersection of their gaze with the destination location on that plane. OrthoGaze can be applied not only with eye gaze and head gaze, but also with joysticks and other gesture raycasting methods.

Results of a user study showed that for aiming tasks in VR over a sustained period of time, eye gaze can outperform head gaze for accuracy, especially for distant targets. Results also suggest that OrthoGaze works well for hands-free 3D manipulation. Compared to manipulation with a hand-held controller, eye gaze-based control was able to achieve approximately 78% performance and head gaze-based control achieved 96%. Additionally, subjective results suggest that using both head and eye gaze for sustained 3D manipulation tasks can comprehensively cause more fatigue than a controller. It is expected that this method will promote new research on eye gaze manipulation, the development of efficient rotation and scaling functions, and extensions to optical see-through HMDs.

Chapter 4

Gaze-based Rotation

4.1 Introduction

In recent years, head-mounted displays (HMDs) have made an increasing appearance as a consumer product. While various interaction methods are available such as controllers, gestures, and voice control, users still lack a private, hands-free method for manipulating virtual objects in 3D space. Tasks such as annotation placement, 3D modeling, and CAD simulation often require translations, rotations, and scaling operations. While hand or controller gestures are the gold standard for these manipulations, users may have their hands occupied by other tasks, feel embarrassed when performing these actions in public, or have handicaps that prevent the use of controllers or certain gestures. As such, many users need access to completely hands-free methods for performing these tasks in AR, VR, and other virtual object manipulation tasks.

To assist with certain actions, eye and head gaze have been seen as an unobtrusive alternative means of selecting objects in 2D and 3D user interfaces (Piumsomboon et al. (2017); Pfeuffer et al. (2017); Vidal et al. (2013a)). As eye gaze can serve as a quick on-screen pointing cursor (Ware and Mikaelian (1986)), it has often been coupled with other means of manipulation to improve accuracy and efficacy. For example, users can use the eye gaze to quickly point at target objects, and then perform a further manipulation through hand gestures (Stellmach and Dachselt (2013); Song et al. (2014); Pfeuffer et al. (2017)). However, manipulations using gaze have traditionally been very difficult due to issues like the Midas Touch problem (Jacob (1995)) since gaze is coupled to a target of interest during the action. Controllers can help solve this problem since the hands and subsequent pointing are decoupled from gaze during interaction, and consequently almost all gaze-based research uses gaze as a support mechanism rather than primary interaction. Moreover, methods that can exclusively use gaze for manipulations are scarce. One example of some progress in this direction is the OrthoGaze (Liu et al. (2020)), which explored the use of eye and head gaze for object translations. However, the method focused on position adjustment, and could not handle more complex

tasks like rotating or scaling an object. Chen et al. (1988) compared a sphere-mapping rotation method with bar-based rotation methods using a mouse on a 2D monitor. While their study gives thorough insights on the design of rotation interfaces with 2D input modalities, it is difficult to directly apply their results to eye-based interaction in a 3D scenario due to the difference in the control methods.

This work takes a step towards completely hands-free manipulation and explore three different techniques for gaze-based object rotations. The first two techniques, RotBar and RotPlane, allow the user to select rotations around one world axis at a time, and they include two unique visualizations. A third technique, RotBall, is also designed that draws inspiration from the traditional arcball manipulation scheme (Shoemake (1992)), though this design is significantly adapted to work with eye gaze. All three methods for rotations went through several iterations of testing and redesign since the nature of gaze interactions are much more confined than hand-eye coupled interactions like controllers or gestures.

A user experiment was conducted to compare the three methods in terms of speed, accuracy, and usability. The experiment included orientation tasks requiring both single-axis and multi-axis alignments. Quantitative results showed that users could perform single-axis orientation with RotBar and RotPlane significantly faster and more accurate than RotBall. On the other hand for multi-axis orientation tasks, RotBall significantly outperformed RotBar and RotPlane in speed and accuracy. Results also revealed that all three methods allowed users to effectively align the object with a mismatched angle less than 6 degrees for both single-axis and multi-axis rotations, and there was no significant difference in terms of the alignment accuracy between the three methods. Results of subjective evaluation suggested that there was no specific tendency in the preference among the three methods.

The contributions of this work are summarized as:

- Three different methods are implemented that improve upon existing work and redesigned them to specifically address the needs of those who need eye-only control to rotate virtual objects.
- A user study is conducted to compare these methods in different situations to determine their suitability for simple and complex object manipulations. The results show that RotBar and RotPlane are more suitable for simple rotations and RotBall is more suitable for complicated manipulations.
- From the results of the user study, several observations were obtained

that need to be considered by future interfaces for gaze-based manipulation.

4.2 Related Work

The research mostly related to this work includes user interfaces that have been designed for manipulating the rotation of virtual objects.

4.2.1 Object Orientation

The function of rotating a virtual object is a fundamental element in virtual environment design, as inspecting an object from an appropriate view point is usually a precursor to further manipulation. Over decades, various devices and methods have been explored that target interactive 3-DoF orientation of virtual objects. Traditionally, 3D rotation in AR and VR have taken advantage of additional devices or hand gestures to give users a more intuitive means of controlling virtual objects. In 2D interfaces, different viewpoints of the object and control bars where users can control 2 DoF at a time have been widely used (LaViola Jr et al. (2017)).

In another example, Chen et al. (1988) compared bar based rotation interfaces with a *Virtual Sphere* visualization that encapsulates the target object into a sphere and maps mouse strokes to "rolling" of the sphere. They also considered a second sphere rotation technique called *XY+Z* that allowed users to continuously control the rotation around the X and Y axes as the user swipes across the sphere, and the rotation around the Z axis by pressing and moving the mouse around the sphere. They found that the slider controls performed faster for simple rotations around a single axis but were slower for more complex manipulations. They also found that most participants preferred using the Virtual Sphere metaphor over the sliders. The Arcball is a method similar to the Virtual Sphere, except that the implementation of rotation is based on quaternion curves (Shoemake (1985, 1992)). Katzakis et al. (2013) evaluated the effects of drawing a sphere around the manipulated object, a method called Arcball-3D, and selection of the rotation point directly through ray-casting, called Meshgrab, for ray-based object manipulation. The rotation was calculated similar to Chen et al. (1988)'s *XY+Z* technique. They found that users could rotate objects faster with Arcball-3D and generally preferred it over Meshgrab.

While the exploration of this work bears some similarity to the work of Chen et al. (1988), their work presented the information to users at a single

location and the rotation controls were aligned with the user's view. While this may be an option on a monitor, on HMDs users can move their voluntarily or involuntarily thus shifting the controls. Thus in this work the rotation of per-axis methods is aligned with axes of the world coordinate system to allow users to consistently control the rotation of the object. Utilizing gaze also presents several additional challenges over a desktop mouse, gaze results in more noisy movements due to jittering and tracking errors. Furthermore, while bars would allow users to rotate the object around 360° Virtual Sphere limits this rotation to at most 180°. While triggering the rotation multiple times may not pose a problem with a mouse, when utilizing eye gaze the selection mechanism may create significant problems. Fixations over a given time period have been widely used to avoid the Midas Touch problem of unintended activations. This could significantly increase the time required to rotate the object when using *Virtual Sphere*. Finally, visual confirmation of the intended rotation with the object's current rotation may cause unintentional rotations. This work thus introduces a modification of the bar controls used by Chen et al. (1988) that would allow users more freedom of positioning their gaze while performing the rotation.

4.2.2 Eye Tracking and Control

One primary advantage of eye-based interaction is that using eye movements requires less muscle compared to performing body movements, which contributes to quick pointing and reducing physical fatigue from long-term usage of the device. For example, Räihä and Špakov (2009) showed that gaze can be used to more quickly select between cursors distributed over multiple screens to increase performance. Additionally, the angular speed of travel of conscious eye movements much faster compared to the hand or body. This is beneficial in terms of performance, but can also be a challenge when trying to achieve eye-based interaction that allows for high degrees of freedom, e.g. moving and rotating virtual objects in a three-dimensional space.

Though less related to object manipulation, Piumsomboon et al. (2017) designed several methods to disambiguate between 3D object selection. These methods, Duo-reticles, Nod-and-Roll, and Radial Pursuit, provide additional methods for selection outside of the traditional gaze-dwell mechanism. Moreover, these methods have a closer relationship with objects in the environment that might be selected for the purposes of augmented or mixed reality. A "Pursuit", the concept originally developed by Vidal et al. (2013b) also make use of a similar mechanism for making selections by tracking the eye's smooth

pursuit of an object and detecting its corresponding trajectory. This allows for a more expressive mode of interaction and location independence since pursuits can have different shapes.

The previous chapter investigated the usability of gaze-only positioning for translations, and found that plane-based interfaces could assist with three-dimensional positioning via eye gaze. One of the methods for performing rotations in this work draws from ideas presented in that plane-based interaction. However, the work is limited to object positioning and the plane-based design is not directly applicable to object orientation due to several limitations in its functionality and interface design. For example, it is difficult for the user to secure a good intersection with the active planes, assuming the user is not moving the object. Thus, further investigation is needed regarding gaze-only object orientation.

4.2.3 Further Motivation

It is not hard to imagine that there exist practical use cases of HMDs where controller-based devices or extensive body movements are not available, e.g. in a crowded public space or for disabled users. Thus options are necessary that allow for completely hands-free interaction with HMDs.

As a step towards this goal, this work is dedicated to handling hands-free object orientation. While large number of methods exist that effectively supports orientation of virtual objects, there is few that works in a hands-free manner. To tackle this issue, this work implemented several prototypes of gaze-based user interfaces intended for manipulating the rotation of virtual objects on HMDs, and investigates the usability of such interfaces through thorough user studies.

4.3 Interaction Methods

This work presents and tests three methods suitable for gaze-based rotation, denoted as RotBar, RotPlane, and RotBall. Each takes advantage of a different mechanism and visualization for performing rotations. This section also describes the adaptations that were necessary to ensure these were usable with eye gaze.

4.3.1 Initial Axis Selection

The first two rotation methods are preceded by an axis selection phase, as shown on the leftmost image in Fig. 4.1. The visualization for this selection includes a set of typical coordinate system axes included in programs like Unity[1] or Blender[2], but the ends of these axes are capped with a circular arrow icon. This 3D visualization helps the user understand the axis of rotation more effectively since he or she has a perspective view of these arrows. Through initial tests where only the axes lines were included, several initial testers mentioned that this was unclear since the lines were essentially 2D. These axes can be selected by any eye based method such as dwell or blink based selections. Upon selecting one of the three axes, the user can then utilize RotBar or RotPlane to complete the rotation. Note that these axes disappear once the user has engaged any of the three rotation methods.

[1]https://unity.com/
[2]https://www.blender.org/

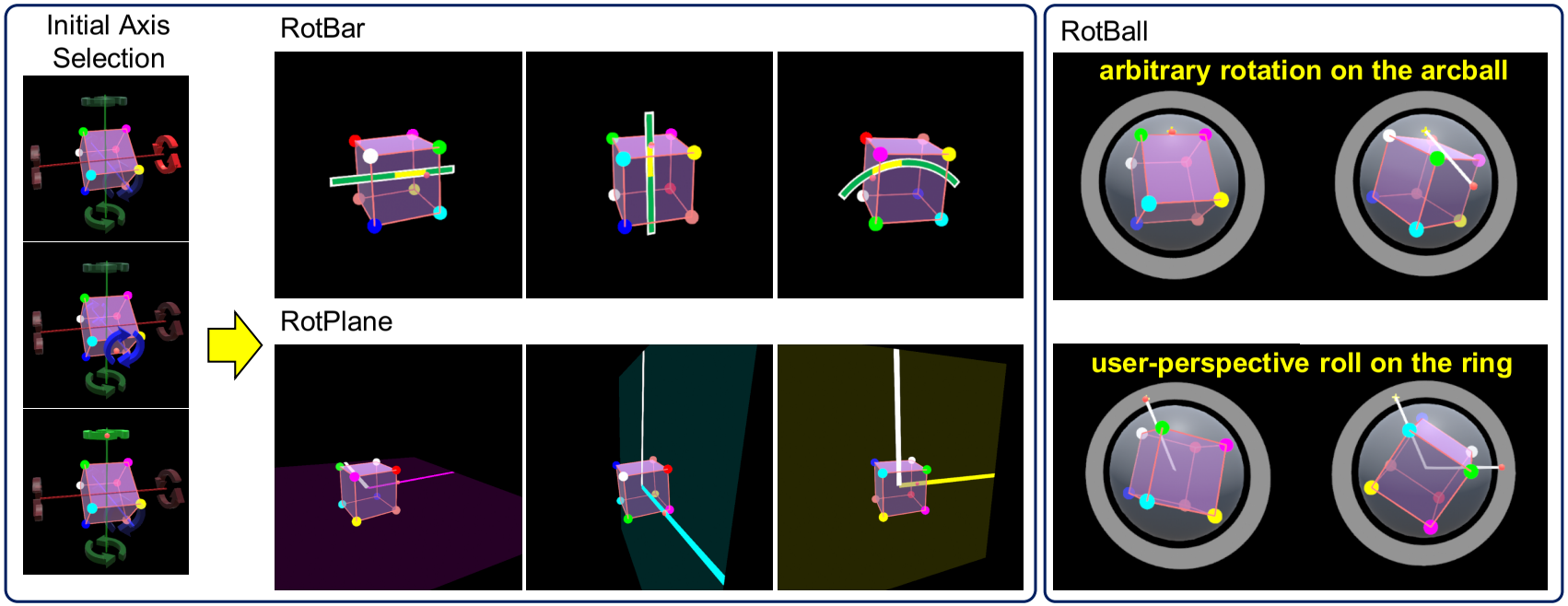


Figure 4.1: Figures showing the three interaction methods. On the left are two per-axis rotation methods: RotBar and RotPlane. A commonality between the per-axis methods is that users first select one world axis (very left of the figure) to enter a rotation mode. RotBar, shown in the upper left row, maps a 360-degree rotation to a bar for each rotation axis where, the center of the bar represents the starting orientation. RotPlane, shown in the bottom left row, calculates a rotation based on the angular offset of where the eye gaze intersects a plane relative to the white bar on the same plane. On the right is RotBall, which is a method composed of a traditional arcball with a surrounding ring that enables user-perspective rolling rotations. In RotBall, users perform a two-step action to apply a rotation, which consists of setting an initial pin and then conducting a rotation based on the new gaze point and that pin. The resulting rotation is calculated as the arc eye gaze trajectory from the first pin (denoted by the line from the yellow plus to the red dot in the upper right row). For the ring, a rotation is calculated as the angular offset from the first pin based on the central point of the ball (denoted by the two white lines attached to the yellow plus and red dot in the lower right row).

4.3.2 RotBar

The first method was RotBar, which is a method that enables per-axis rotation. It is designed to minimize the amount of movement users must do to manipulate the object. It makes use of three "bars" that travel along and correspond to the selected axis of rotation. These essentially function as a visual gauge to determine the degree of rotation, as shown in Fig. 4.1. Rotations are mapped so that the one side of the bar corresponds to a 180-degree rotation in one direction and the other side corresponds to 180 degrees in the opposite direction.The centers of the three bars shown in the top left row of Fig. 4.1 represent the initial orientation of the object during each rotation action. As the user's gaze proceeds along each of these bars, the gauge turns from green to yellow, and the object rotates synchronously with the gaze change.

Thus, a rotation r [rad] on each axis is calculated based on a 1-dimensional gaze offset d from the central white bar as:

$$r = 2\pi d, \tag{4.1}$$

where d is given corresponding to the rotation axis as:

$$d = \begin{cases} \dfrac{gaze_x - bar_x}{\mathrm{L_h}}, & \text{yaw and roll} \\ \\ \dfrac{gaze_y - bar_y}{\mathrm{L_v}}, & \text{pitch} \end{cases} \tag{4.2}$$

where $gaze_{x,y,z}$ denotes the gaze position, $bar_{x,y,z}$ denotes the position of the bar center, and $\mathrm{L_h}/\mathrm{L_v}$ denote the horizontal/vertical length of the gauge, respectively. In short, the gauge linearly maps a full 360-degree rotation for any axis to the offset from the corresponding central bar to the gaze point.

When one of the axes appears almost parallel to the user's viewpoint, e.g., $<10^\circ$ offset, it would be very difficult for the user to control the rotation accurately. To prevent this, the visualization is switched from a bar to an arc around the axis. In Fig. 4.1 this applies to the cube's roll.

4.3.3 RotPlane

RotPlane makes use of three orthogonal planes to handle the per-axis rotation, as shown in Fig. 4.1. Here the user's gaze functions like a handle that is rotated from its original location (white bar) to the target location (the user's gaze) on a plane around the corresponding axis. Assuming rotation r [rad] is clockwise at each axis, it can be acquired using the same equation as Eq. (4.1), but

in this case the offset d is given by a planar angular offset between from the white bar to the gaze point as:

$$d = \frac{\angle(\hat{bar}, \hat{gaze})}{2\pi} \tag{4.3}$$

where $\angle(\hat{a}, \hat{b})$ denotes a directional angle from vector $\hat{a}$ to $\hat{b}$ resulting in $[0, 2\pi]$, $\hat{gaze}$ denotes the vector from the center of the plane to the gaze point, and $\hat{bar}$ denotes the vector from the center to the tip of the bar. The larger area where users can position their gaze and the alignment of the rotation indicator with the user's gaze could help users keep their focus on the object they manipulate instead of the control elements.

4.3.4 RotBall

Previous work showed that sphere based control methods can be efficient in controlling an object's orientation. To investigate the applicability of these findings to gaze based manipulation, RotBall was designed that combines a traditional arcball mechanism with a surrounding ring to support rolling rotations that would otherwise be difficult due to the user's perspective. A sphere is visualized around the object and users can rotate the object by interacting with the sphere. As such, it is similar to Chen et al. (1988)'s *Virtual Sphere* and Katzakis et al. (2013)'s *Arcball-3D*. Since the arcball mechanism does not require a single-axis selection, there is no axis selection phase. Instead, the user performs a 2-step command to apply a rotation: setting an anchor point and then acquiring a rotation based on the spherical gaze trajectory from the anchor point. This is similar to the placement and release of a finger on a ball mouse. The initial placement of the user's finger sets the initial point to drag, then dragging his or her finger will rotate the ball in-place, and finally the removal of the finger will disengage the rotation of the ball.

In general, a rotation r [rad] on the RotBall is calculated as

$$r = \angle(\hat{anchor}, \hat{gaze}) \tag{4.4}$$

where $\hat{gaze}$ denotes the vector from the center of the RotBall to the second gaze point, and $\hat{anchor}$ denotes the vector from the center to the first anchor.

4.4 Experiment

This section introduces the details of a user experiment that was conducted to validate the usability and user experience of each rotation method. In the

experiment, participants were asked to use RotBar, RotPlane and RotBall to complete a set of orientation tasks (docking tasks) in an HMD. For all three methods, participants used eye gaze to rotate cubes as closely as possible to a target rotation, represented by a semi-transparent target cube as shown in Figure 4.2. Both the target cube and cube to rotate contained 8 colored points, one for each corner, to help the user understand the orientation and confirm that the task was completed correctly. Though gaze dwell, blink, or other eye-based selection methods are compatible with these rotation methods, participants were asked to make and confirm selections using the trigger and touch pad buttons on an HTC Vive controller. Based on previous work (Liu et al. (2020)), the user's familiarity with eye-based selection methods could bias the performance and user experience of a 3D manipulation method. To exclude the effect of bias due to selection methods and focus on the performance of rotating the object itself, this work opted for the controller as it presented a familiar interface that all users could operate quickly without extensive training. This also helped avoid the Midas touch problem Jacob (1995), which is a common issue for eye gaze-based selections but was not the target of this experiment.

4.4.1 Hypotheses

The main purpose of the experiment is to evaluate how quickly and accurately users would perform with each method for eye gaze-based orientation tasks. Based on the previous findings of Chen et al. (1988), it was expected that for single-axis orientation tasks, axis-based methods could outperform arcball-based methods in terms of speed, and the opposite for multi-axis tasks. While RotBar and RotPlane map a full 360-degree rotation in a specified field of view, RotBall has the least sensitivity and was expected to be most suitable for small alignments. In addition, compared to Rotbar, RotPlane occupies a larger field of view and could potentially be easier to perform slight alignments. Thus it was expected that RotPlane will overall outperform RotBar. To summarize, the following hypotheses were formulated:

H1a Participants will complete single-axis orientation tasks most quickly with RotPlane, followed by RotBar, and then RotBall.

H1b Participants will complete the multi-axis orientation tasks most quickly with RotBall, followed by RotPlane, and then RotBar.

H2 Participants will complete the orientation tasks most accurately with RotBall, second with RotPlane, and third with RotBar.

4.4.2 Setup and Participants

In total, 11 students and researchers were recruited from multiple universities, 6 male and 5 female, ranging in age from 22 to 33 (avg. 27.2, stdev. 3.7). Of these participants, 6 were wearing prescription glasses with the HMD during the experiment. 5 of them had no experience in eye tracking and eye gaze-based human-computer interaction, 2 had less than five times in total, while the remaining 4 had more experience (more than five times in total) before this experiment. 4 of them ran this experiment remotely on their personal home or office in order to mitigate the transmission of infectious diseases during in-person experiments. The in-person experiment ran on a desktop computer with an Intel Xeon E5-2690 CPU and an Nvidia GeForce GTX 970 GPU at an average frame rate of 40 frames per second. For the HMD, all participants used an HTC Vive Pro Eye that has integrated eye tracking cameras and provides relatively stable eye gaze data. For participants running the experiment remotely, it was ensured via a remote connection that all participants could run the eye tracking, rotation interfaces, and experiments without any issues. When the experiment was conducted remotely, an experiment conductor supervised the remote participant throughout the experiment to ensure that the external conditions matched that of in-person participants as closely as possible. Participants who took part in the experiment in-person received a gift card worth approximately 5 USD as remuneration.

The virtual environment was created using Unity 2018.3.2f1 and contained all of the rotation methods and experiment tasks. The eye tracking data was run and recorded in real-time through the Vive Sranipal runtime version 1.1.2.0. A smoothing filter was applied that aggregated and averaged the eye gaze data for the last 5 frames. The eye gaze point was visualized as a red dot only when the eye gaze intersected with interactive objects and interfaces. The statistical evaluation of the experimental data was processed using R 3.6.2 [3] and coin 1.3-1 [4].

4.4.3 Procedure

When participants entered the experiment environment, they first received an introduction to the experiment tasks and an explanation of each rotation method. Second, they read and signed a consent form that explained the details and risks of the experiment. During the experiment participants were asked to remain seated on a swivel chair and not to stand up or move around.

[3] https://www.r-project.org/
[4] http://coin.r-forge.r-project.org/

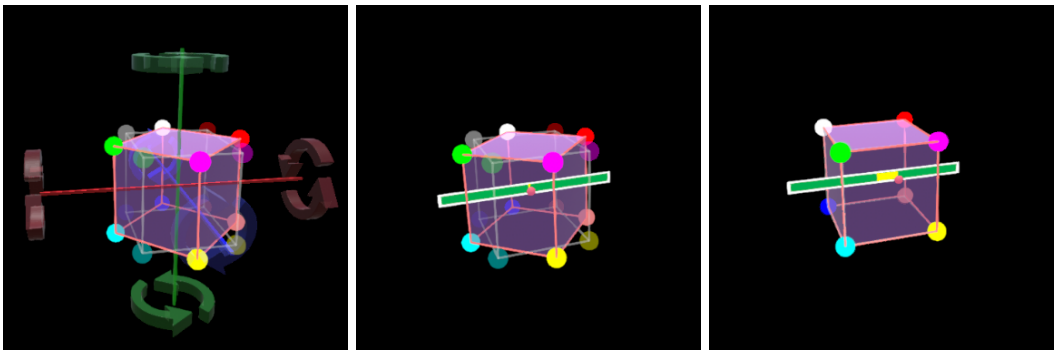


Figure 4.2: Sample images showing the orientation task in the user study. (a) The default state where the two cubes are of different orientations. The user selects an axis in this phase. (b) After choosing an axis, the rotation mode (in this case RotBar) is engaged. (c) Image showing that the two cubes have been aligned, after which the user can confirm the rotation and end the trial.

Local body movements were not physically restricted, so participants could rotate their chair if necessary. The participant's neck, head, or seated body movements were also not restricted to ensure that the participants could view the tasks in a natural manner and perform natural eye movements supported by head movements.

Participants ran the Vive Pro Eye integrated eye tracking calibration at the very beginning of the experiment. The tracked gaze point was also shown to participants during the experiment to ensure that the tracking accuracy was adequate. Participants were able to pause the trial and rerun the eye tracking calibration if they felt the eye tracking had become inaccurate (e.g. due to drift) during the experiment.

Before entering the formal trials, participants had a training session where they could practice each method until they felt comfortable, and could take a break between each trial if needed. Participants had to finish all trials with one method, and then proceeded to the next. The order of the methods was counterbalanced using a Latin square to alleviate ordering effects. After finishing all trials, participants completed a custom survey (A.2) which is partially based on the System Usability Scale (SUS) (Brooke (1996)) for each method.

Overall, the experiment took from 60 to 120 minutes, including the consent process and surveys. The procedure of the experiment was approved by the institutional review board of Osaka University Review Board.

4.4.4 Tasks and Conditions

As shown in Fig. 4.2, participants first see two cubes positioned in front of them, a pink cube that is rotatable, and a white-framed cube that has a different (target) orientation. The task is to rotate the pink cube to match the target orientation using the 8 colored dots affixed to each corner. The starting view point was positioned at (0, 1, 0) unity units (meters) and faced the front direction: (0, 1, 1) meters. The orientation tasks for each method included two sets of 3 single-axis and 1 multi-axis matching tasks, $(1/4\pi, 0, 0)$, $(0, 1/4\pi, 0)$, $(0, 0, 1/4\pi)$, and $(1/4\pi, 1/4\pi, 1/4\pi)$ [rad] in the world coordinate system, ordered from single-axis to multi-axis. The two sets appeared at two different locations: (0.75, 0.3, 2) meters and (-0.75, 0.3, 2) meters. This resulted in 8 trials per method, with 6 single-axis and 2 multi-axis, for a total of 24 trials for the experiment. The order of the methods was counterbalanced using a Latin square to alleviate ordering effects.

As mentioned previously, participants used 2 buttons of an HTC Vive controller: the trigger button for making selections (e.g. selecting a rotation axis and fixating a rotation), and the touchpad button for making confirmations (e.g. confirming that they had finished a trial and proceeding to the next trial). Additionally, the controller had no aiming or laser function, as participants had to exclusively use their eye gaze to utilize each method to rotate the object.

As the priority goal of each task, participants were asked to match the orientation as accurately as possible. Although the time taken for each trial were being recorded, there was no time limit set. The timer of each trial would not start until the participant made the first selection (trigger button), giving participants sufficient time to observe the target orientation at the very beginning of each trial without needing to pay attention to a timer. After participants rotated the object and decided that it was correctly aligned with the target, or that they could not align it more accurately, they pressed the touchpad button to conclude the trial and proceed to the next one. Note that participants could not perform the final confirmation while in the rotation mode, which means participants had to affix the in-progress rotation prior to concluding the trial. Between each trial, participants could take breaks without any time restrictions and proceed to the next trial by pressing the touchpad button when ready. Participants had full control of the trial start time so that they had ample time to understand the target orientation. Otherwise this mental effort might have biased the rotation times.

For evaluating the experimental results quantitatively, the following metrics were recorded and calculated:

Completion time: Completion time is the time between the first selection and the confirmation of the alignment by the participant. For a more objective comparison, the completion time of single-axis and multi-axis tasks is shown separately since the multi-axis tasks were more complicated. Note that there was no time limit set for each trial, which means the completion time could become either very short or extremely long depending on how the participant defined a "good match". Although limiting the time to one minute could have allowed for more trials, unlimited time was opted for as this would allow to analyze both time and accuracy without any trials that were cut-off mid-rotation.

Misalignment: Misalignment is defined as the mismatched angle in [rad] by calculating the angular offset of the rotated object from the target orientation for each trial. This essentially gives us an idea of the accuracy with which participants were able to match rotations. These results are also shown separately for single-axis and multi-axis tasks.

Number of selections: The number of selections included the total number of trigger presses from the start of the first rotation to the end of the trial. This would give us an idea of how many interactions (dwells in the case of complete eye-control) were necessary for each method.

4.4.5 Results

This section describes both the quantitative (speed, misalignment, and number of selection) and qualitative (subjective scoring from the SUS questionnaire) results of the experiment. The data were verified if they met with the requirement of ANOVA by performing Anderson-Darling normality tests and Mauchly's W sphericity tests. A threshold of $p = 0.05$ was used to determine statistical significance. $r = Z/\sqrt{N}$ is reported as the effect size for post-hoc Wilcoxon signed rank test results, where Z is the statistical value and N is the total sample size (Rosenthal (1994)).

Fig. 4.3 (a) shows the completion time of both single-axis and multi-axis tasks. One-way ANOVA with repeated measures yielded significant variation among all three methods for both conditions (($F(2, 185) = 13.90, p < 0.001$) for single-axis tasks and ($F(2, 53) = 17.40, p < 0.001$) for multi-axis tasks).

For single-axis tasks, a post-hoc Tukey test showed significant difference in completion time between RotBar and RotBall ($p < 0.001$), and between RotPlane and RotBall ($p < 0.001$). Average times taken for single axis tasks were 41.78, 39.06, and 90.85 seconds for RotBar, RotPlane, and RotBall, respectively. RotBall took more than double the time of the other methods.

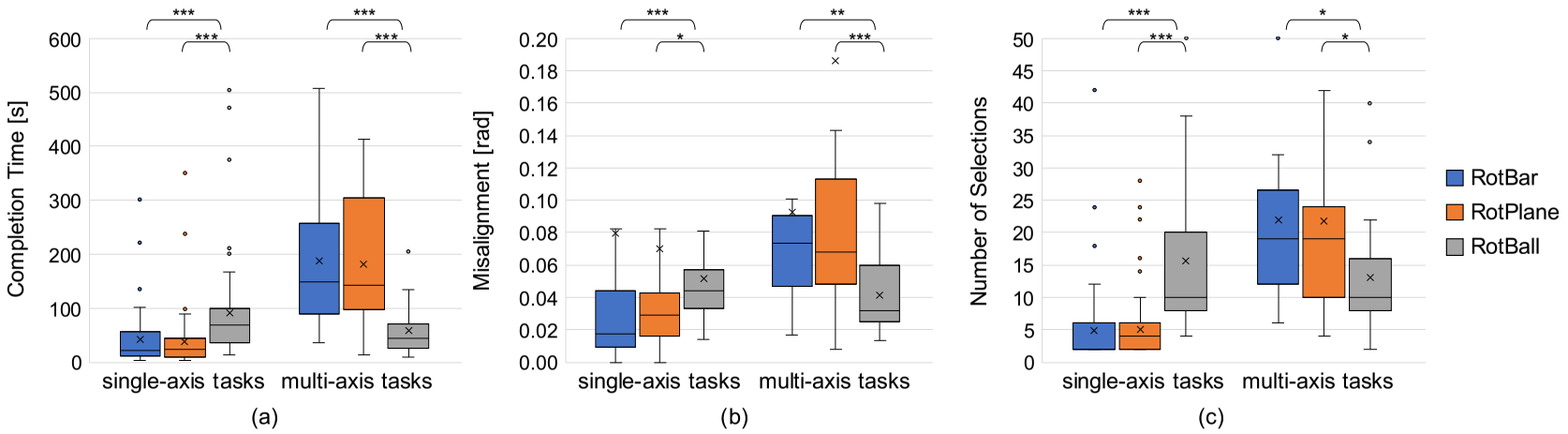


Figure 4.3: Boxplots showing (a) completion time for both single-axis and multi-axis tasks, (b) misalignments for both single-axis and multi-axis tasks with outliers excluded for clarity, and (c) number of selections (trigger presses) required to complete a trial. (***: $p < 0.001$, **: $p < 0.01$, *: $p < 0.05$)

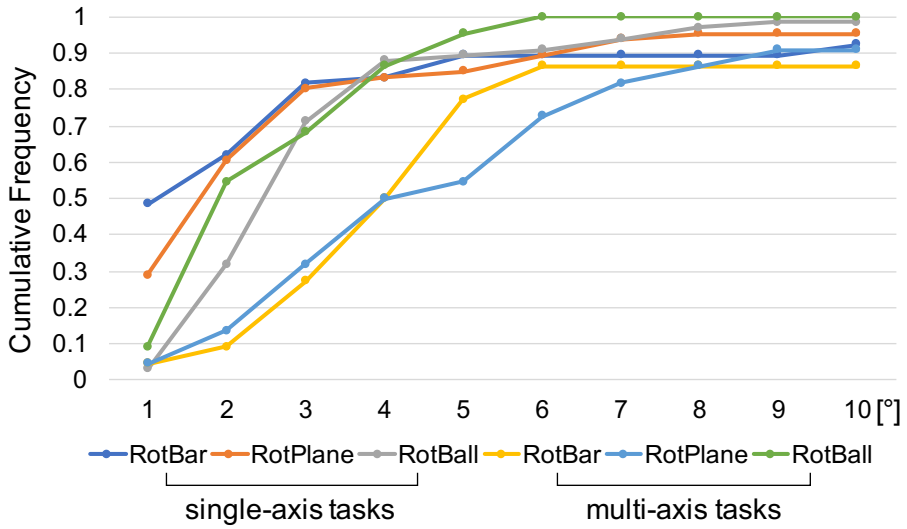


Figure 4.4: A graph showing the cumulative frequency curve of the matching results of each method for box single-axis and multi-axis tasks, within a range of 10-degree misalignments devided by every 1 degree.

For multi-axis tasks, a post-hoc Tukey test showed significant difference in completion time between RotBar and RotBall ($p < 0.001$), and between RotPlane and RotBall ($p < 0.001$). Average times taken for multiple axis tasks were 187.13, 181.78, and 54.46 seconds for RotBar, RotPlane, and RotBall, respectively. In contrast to single access tasks, RotBall took less than a third of the time of the other two methods.

Fig. 4.3 (b) shows the results of the misalignment of each method for both single-axis and multi-axis tasks. For single-axis tasks, Friedman's tests showed statistical significance between the different methods ($\chi^2(2) = 25.41, p < 0.001$). A post-hoc Wilcoxon signed rank test with Bonferroni correction showed significant differences between RotBar and RotBall ($Z = 3.62, r = 0.32, p < 0.001$), and between RotPlane and RotBall ($Z = 2.75, r = 0.24, p < 0.01$). For multi-axis tasks, Friedman's tests showed statistical significance between the different methods ($\chi^2(2) = 13.73, p < 0.01$). A post-hoc Wilcoxon signed rank test with Bonferroni correction showed significant differences between RotBar and RotBall ($Z = 3.04, r = 0.46, p < 0.01$), and between RotPlane and RotBall ($Z = 3.33, r = 0.50, p < 0.001$).

In terms of accuracy, a cumulative frequency curve is also shown in Fig. 4.4 covering the percentages of how each method performed the matching for each task. For single-axis tasks, all three methods received an over 70% matching with the misalignment less than 3 degrees. For multi-axis tasks, RotBall

received a higher rate of nearly 70% for matching with the misalignment of less than 3 degrees, while RotBar and RotPlane reached over 70% matching of misalignments less than 6 degrees.

Fig. 4.3 (c) shows the number of selections performed for accomplishing each task. One-way ANOVA with repeated measures yielded significant variation among all three methods for both conditions (($F(2, 185) = 35.94, p < 0.001$) for single-axis tasks and ($F(2, 53) = 4.70, p < 0.05$) for multi-axis tasks). For single-axis tasks, a post-hoc Tukey test showed significant difference between RotBar and RotBall ($p < 0.001$), and between RotPlane and RotBall ($p < 0.001$). For multi-axis tasks, a post-hoc Tukey test showed significant difference between RotBar and RotBall ($p < 0.05$), and between RotPlane and RotBall ($p < 0.05$).

In general, quantitative results revealed that for single-axis tasks, RotBar and RotPlane outperformed RotBall in both speed and accuracy, while for multi-axis tasks, RotBall oppositely was faster and more accurate than RotBar and RotPlane.

Fig. 4.5 (a) shows the results of the SUS survey for each method. Friedman's tests showed no significant difference between methods for each survey item. The system usability (SU) scoring was also calculated and shown in Fig. 4.5 (b) using Brooke's standard scoring method (Brooke (1996)). All three methods received a medium scoring in average, 60.00 for RotBar, 59.55 for RotPlane and 52.05 for RotBall. A Friedman's test showed no significant difference in the scoring between each method. Participants were also asked about their most preferred method. Among 11 participants, 4 voted for RotBar, 4 voted for RotPlane, and the remaining 3 voted for RotBall. In general, there was neither significant difference in the usability of the three methods, nor significant trend in the individual preference found.

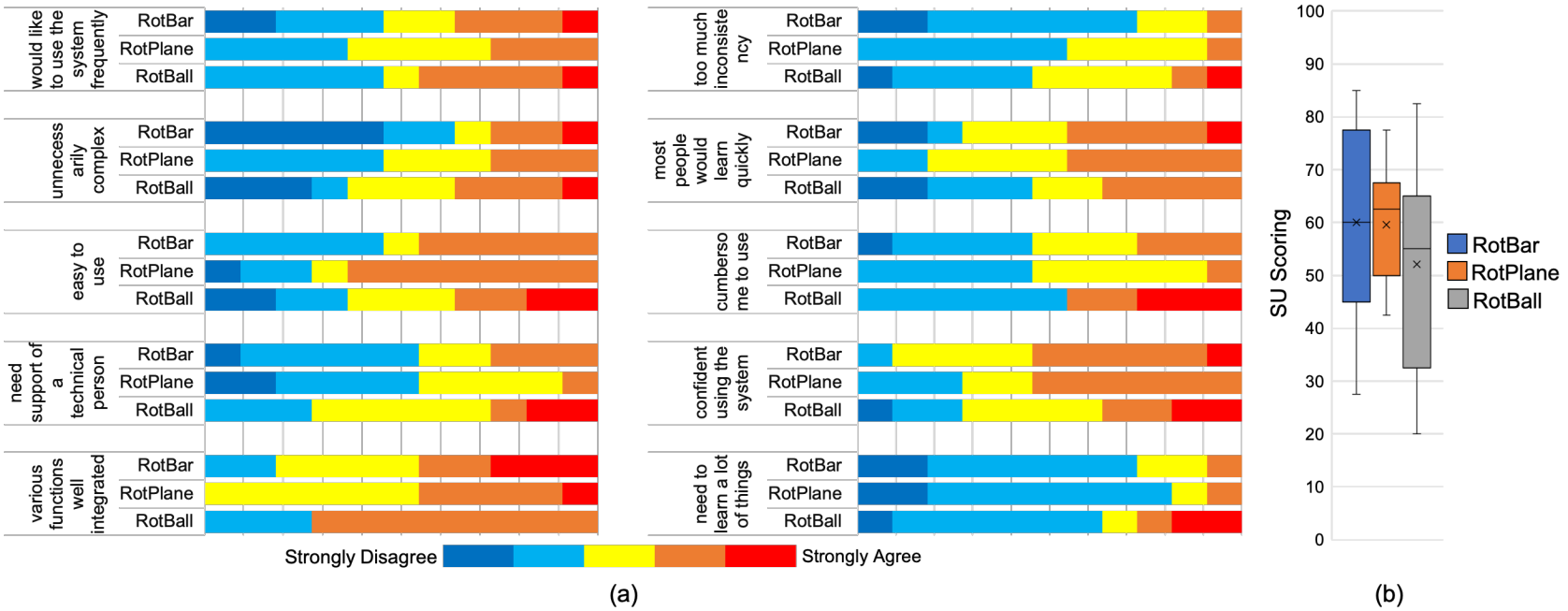


Figure 4.5: (a) Charts showing the overview of 5-point Likert scale ratings of each system usability survey item. (b) A boxplot showing the average system usability (SU) score of each method from all parcitpants, as calculated by Brooke's standard scoring method (Brooke (1996)).

4.5 Discussion

Experimental results revealed that RotBar and RotPlane were completely opposite of RotBall when comparing single- and multi- axis tasks in terms of speed, as outlined in Figure 4.3 (a). These results are in line with previous findings by Chen et al. (1988) and partially support **H1a** and **H1b**, and suggests that it may be better to use a combination of the two types of method during 3D modeling or other object rotation tasks. When more complex rotation tasks are necessary, RotBall should be used, but for smaller or single-axis rotations RotBar or RotPlane would be faster. This same tendency was present for the total number of controller presses, as shown in Fig. 4.3 (c).

Experimental results regarding accuracy suggested that for single-axis tasks, the per-axis methods were significantly more accurate than the arcball-based method. While for multi-axis tasks, the arcball-based method could significantly outperform the per-axis methods in accuracy. This result partially supports **H2**. It is suspected that because the participants were given unlimited time to complete a rotation, final accuracy ended up being more of a function of personal preference rather than a characteristic of a specific rotation method. Results might have been different if the trial time was capped, but this would not be representative of real world manipulation tasks such as 3D modeling or design in which users can take their time.

Although it was expected that RotPlane would help users to align the two cubes as users could adjust the displacement of their gaze from the center, thus keeping the two cubes in focus at all times, the results and comments from the participants showed that this was not the case. It is believed that the main reason for that was that participants relied on the corners to verify the quality of the alignment. As the initial orientation of the cube would then be aligned with the user's gaze, participants could not always rely on the point they were fixated on to determine the alignment quality.

4.5.1 Design Implications

Although participants could align the cube with its target using all 3 methods and did rate them similarly, all methods were rated lower than the median score of previous studies that utilized the SUS (Bangor et al. (2009)). In the following, some observations are discussed that could explain this result and how the methods could be improved.

For RotBar and RotPlane, a 360-degree rotation was mapped with equivalent rotation speed to each axis for achieving a maximum rotation range. Subjective comments gave that in the cases of small adjustments, the rota-

tion speed was too fast compared with the eye gaze movements, e.g., rotating 1 degree was more difficult than rotating 90 degrees as the user had to perform very minute eye movement. Considering that practical rotation tasks often require small adjustments, it is suggested that an improved gaze-based rotation interface should include rotation speed adaptive to the task needs to enable huge fast adjustments and small precise adjustments simultaneously. Nonisomorphic mapping of rotation amounts has been shown to lead to faster rotations without a loss of accuracy (Poupyrev et al. (2000)) and could potentially be applied here as well. A static or adaptive nonisomorphic mapping of gaze locations to the amount of rotation could help users make minute adjustments, but could lead to increased operation time if users have to select the rotation mode again for final adjustments.

An issue was also observed that would affect the results of the orientation tasks and increase users' frustration. As part of the task design in the experiment, participants also needed to check whether the two cubes were well-aligned during rotation. In some cases, e.g., when the corners shifted from a participant's central/paracentral vision into the peripheral vision during the manipulation, trying to match the corners would cause a sudden eye movement and accidentally trigger an unintended rotation. From the author's observation, the issue was likely due to the nature of the matching task, which required the user to view the cube in parallel to the manipulation. This could also explain the difference in our findings from those of Pathmanathan et al. (2020), who collected feedback after a free manipulation task without a defined target object pose. One available way to tackle this issue would be improving the design of the interfaces to focus important information in the central/paracentral vision, such as showing a small copy of the rotated object at the eye gaze position. However, this could lead to clutter in the user's view. Alternatively, a world-in-miniature representation placed in an area that does not overlap with other content could also help user's to more easily verify the rotation without shifting their gaze. This issue is considered to be especially critical for eye gaze-based manipulation interfaces, especially for use cases requiring simultaneous matching and manipulation, which deserves further investigation.

In the experimental implementation, no degree curve for visual guidance was integrated in purpose for all three methods to test how the eye gaze could utilize the three 3D rotation methods in a most fundamental way. However, it is expected that visual guidance such as grid textures could help improve the usability of 3D manipulation interfaces. For eye gaze-based interactions, it is expected that visual guidance could make it easier for users to aim at

specific positions and thus improve the accuracy as well as the stability of the system to some extent. On the other hand, inadequately integrated visual information could also lower the performance, e.g. extremely detailed grids representing every 1-degree rotation on the arcball might instead confuse the user, as the eye might have to continuously filter out information.

As a future step regarding eye gaze-based rotation interfaces, it is worthwhile to explore how different visual guiding textures could affect the performance, and to discover visual guidance optimal for eye gaze-based rotation.

In addition, this work mainly focuses on evaluating how eye gaze will perform on object rotation. Thus we implemented all tested methods with a controller to support selections. On the other hand, it is also crucial to further explore optimal gaze-based selection methods for rotation interfaces. An inappropriate selection method would lead to a poor user experience due to problems such as Midas touch. However, some existing solutions that require additional eye movements, e.g. "Pursuit" (Vidal et al. (2013b)) or "DualGaze" Mohan et al. (2018), are not completely compatible with the rotation interfaces. Unlike selection tasks, rotation tasks ask for sustained and stable focus of attention on the interface, which does not allow extra eye movements during the manipulation. Previous work (Liu et al. (2020)) found that while gaze dwelling could work with gaze-based manipulation interfaces, this was difficult for some users to perform and might have caused frustration during the experiment. Thus, there is still a lack of efficient selection methods optimized for manipulation interfaces, which needs further development and investigation.

Future work also includes the extension of eye gaze-based manipulation methods to scaling operations and full combined 9-DoF translation, rotation, and scaling tasks.

4.6 Chapter Conclusion

This work designed and explored the usability of three different methods, RotBar, RotPlane and RotBall, that are suitable for eye gaze-based rotation of 3D objects. RotBar makes use of three bars to handle the per-axis rotation and acquires a rotation determined by how much the eye gaze positions from the center of the bar. RotPlane enables per-axis rotation using three orthogonal planes and calculates a rotation based on an angular trajectory on the plane. RotBall combines a traditional arcball that allows arbitrary rotations with a surrounding ring that is supportive for user-perspective roll rotation. Experimental results showed that RotBar and RotPlane were faster and more

accurate in performing single-axis rotations, but that RotBall greatly outperformed the other two methods for multi-axis rotations. As none of the methods were clearly preferred by participants the methods could be selected specifically for the task at hand, such as RotBar or RotPlane for rotating an avatar or a virtual box placed on the floor and RotBall for rotating a virtual model that the user wants to look at from all sides.

The interaction design can serve as a significant step towards 3D modeling, editing, and interaction using pure eye gaze, and these methods will be advantageous for individuals that have limited use of their hands or arms during interaction. It is also expected that this work will help pave the way for new eye gaze-based manipulations moving forward.

Chapter 5

Pupillometric Light Modulation

5.1 Introduction

OST-HMDs give us the ability to display digital content over the real world in a user's direct field of view. Ideally, a user should be able to clearly see the augmented information without disturbing information in the real world in the process. For example, when reading an e-mail in a dim room, solid billboard text could occlude the scene and prevent the user's pupils from light-adapting to the natural environment. Mismatched brightness can also cause eye fatigue and significant reductions in performance (Duffy and Chan (2002)).

To prevent this lighting mismatch and reduce the need for constant manual adjustment, digital content should be displayed at the preferred, rather than most visible, brightness in order to obtain balanced lighting conditions in which users can view both scene and display content comfortably. Though some OST-HMDs have built-in functions that adjust the brightness of the display based on the environment (Wong and Mirov (2015)), matching environment brightness or changing text color to maximize visibility is not always the best solution, especially for non-augmentative or user-centric content (Orlosky et al. (2013)). For example, for the same block of text and background, different viewers will prefer different display contrast since each individual's perception and light-adapted pupil size differs. Also, looking at a small text label versus a large browser window at the same luminance will have a very different effect on pupil dilation and user perception. In contrast with algorithms designed to maximize visibility (Gabbard et al. (2007)), light adjustment for practical use needs to handle issues like inadequate light-adaptation of the eye in dim environments and non-linear contrast preferences that vary from user to user.

To address the aforementioned problems, this work proposes a novel method called IntelliPupil that is designed to help automatically adjust the brightness of OST-HMDs (Fig. 5.2). Rather than using a scene camera and matching or maximizing contrast with environment light like other approaches, IntelliPupil uses a combination of pupil size taken from a near-eye camera and light readings from a high dynamic range, small form-factor light sensor. The data

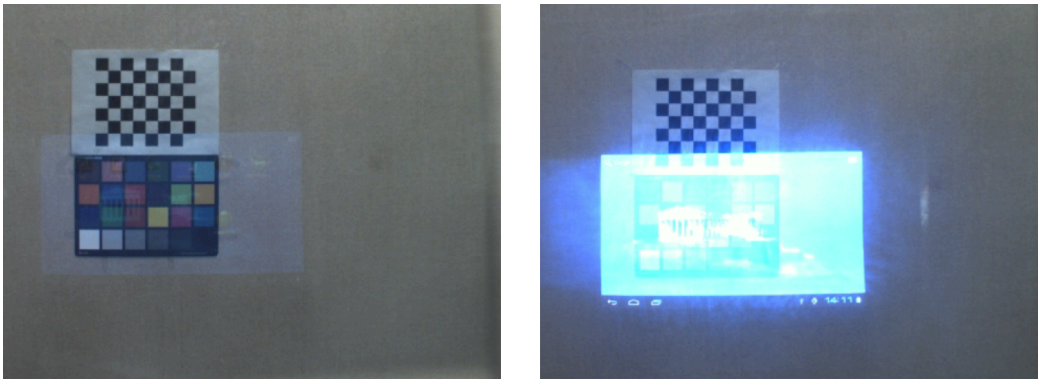


Figure 5.1: Images showing the visibility issues of OST-HMDs caused by mismatched brightness. Left: The HMD is too dim to recognize. Right: The HMD is too bright, preventing the user from properly viewing the real scene.

from these two inputs is then used to train the algorithm, which is able to adjust brightness using a combination of machine learning and filtering for an individual user in any environment in real time. This strategy also accounts for gaze point on both the screen and environment since the pupil inherently adapts to any light passing through to the retina. Several significant discoveries have also been made regarding the correspondences between user preference and pupil size, as well as the pupil's response to virtual lighting.

First, a pilot experiment tested a simplistic pupil-based method in order to get an idea of how effective the pupil itself would be as an input for adjustment. Results show that using only the pupil performs relatively well in maintaining optimal brightness to some extent. However, this approach still has several limitations due to the non-linear nature of user contrast preferences over higher dynamic ranges.

Based on these findings, this work proceeds to create an improved algorithm that uses combinations of user preference, pupil size and environment lighting to make the adjustments. After implementation, several iterations, and refinements of the algorithm, a developed version of IntelliPupil is produced, and a user experiment is conducted to evaluate how well IntelliPupil will perform against current linear model that is implemented for comparison. These algorithms are tested with both virtual text and billboard content, combined with a number of different backgrounds over a high dynamic range to test both the accuracy and practicality of the system. Experimental results show that IntelliPupil outperforms the linear model in most cases, and further analysis reveals that pupil size just as important of a factor as environment light for matching preferred brightness. In short, the primary contributions

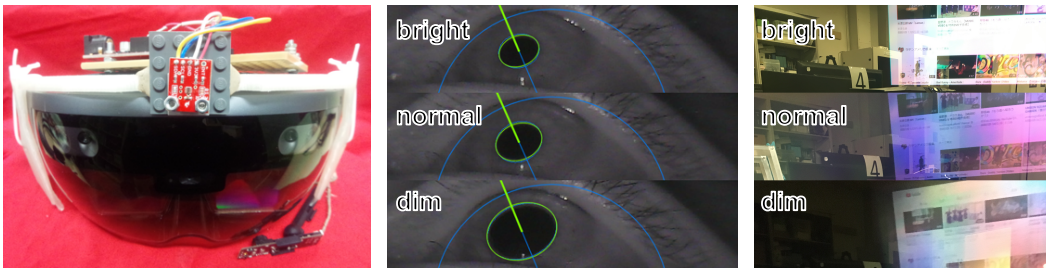


Figure 5.2: Images showing an overview of the IntelliPupil system. Left: The OST-HMD setup used in this work with an eye tracker and a light sensor. Middle: Light-adapted pupil ellipses in bright, mid, and dim light conditions. Right: Algorithmically chosen HMD brightness for corresponding pupil-light data pairs, paused over real backgrounds and taken through the user's view point).

of this work can be summarized as follows:

- This work proposes a novel algorithm, IntelliPupil, that accounts for user preference, pupil size, and environment light to manage display brightness.
- An experiment is conducted to compare IntelliPupil against a linear model. The experimental results reveal the effectiveness of pupil size for HMD light modulation.

The remainders of this chapter present existing literature on pupillometry and the effectiveness of lighting adjustment techniques to improve AR experiences with OST-HMDs, present details of the pilot experiment, describe the algorithm re-design and refinements that are carried out based on the pilot experiment data, present the details of an in-depth experiment to test the performance of the improved algorithm, and finally discuss and conclude the work.

5.2 Related work

Most research relating to this work falls into one of two categories, including 1) eye tracking and pupillometry applications used to control AR content, and 2) methods specific to automating lighting adjustment.

5.2.1 Eye Tracking and Pupillometric Measurement

Eye tracking is a good means of improving the user experience for OST-HMDs since it can provide a variety of valuable information and is easy to implement even on a wearable device.

For example, it has been explored as a means for interaction (Lee et al. (2014)), focus depth estimation (Toyama et al. (2014); Lee et al. (2012)), and calibration (Itoh and Klinker (2014); Plopski et al. (2016)) for OST-HMDs. While most common applications of eye tracking in OST-HMDs remain the estimation of either the eye pose or the user's gaze, the variation of the pupil size also provides information that is essential for generating more compelling computer graphics or world-registered augmentations (Oshima et al. (2016); Rompapas et al. (2017)).

Many studies are already dedicated to the study of how the pupil and eyes are affected by light. For example, an early study by De Groot and Gebhard (1952) came up with a model for determining pupil size in response to luminance. Though the pupil functions similar to the aperture of a camera in that it controls how much light falls onto the retina to adapt to the scene, its size is affected not only by the amount of incoming light, but also by a wide range of factors, such as age (Watson and Yellott (2012)), mental workload (Pfleging et al. (2016)), mental state (Bradley et al. (2008)), and iris color (Winn et al. (1994)). As such, the pupil size plays an essential role when studying the mental state of the user, which can also affect his or her perception of environmental light.

Pupil size has also been used to study effects of watching media on the user and to adapt the luminance of a monitor to present a better viewing experience (Taptagaporn and Saito (1990)). This work follows this idea, but unlike the case of a monitor or an immersive HMD, lighting adjustment of OST-HMDs must take into account that users can view the background lighting as well as the illumination coming from the HMD in the same field of view. It is thus necessary to provide not only a comfortable experience, but also to consider the balance between incoming light sources so as to not prevent viewing of one or the other.

Instead of considering the pupil variation as a fatigue level as Taptagaporn and Saito (1990), this work treats it as a parameter to describe the user's real-time adaptation state to all the perceived light. By referring to it and combining it with the user's preference, it is able for the system to train the algorithm and modulate the brightness/contrast to a more balanced level.

5.2.2 Automated Lighting Adjustment

Automated adjustment of brightness/contrast of the virtual content relative to the background real environment under dynamic lighting is an active research topic in AR for both HMD-based systems (Yamazoe et al. (2009); Mori et al. (2018); Hiroi et al. (2017)) and other displays like projectors (Fujii et al. (2005)).

For an immersive HMD, brightness/contrast adjustment is somewhat straightforward since the display content is all that the user views (Zhao et al. (2015)). In the case of OST-HMDs, however, brightness/contrast adjustment requires a more careful control mechanism. One attempt at solving this problem was by Yamazoe et al. (2009), who investigated algorithms to adjust LCD backlighting and found that a linear method for adjustment outperformed displaying at a middle brightness level for viewing HMD content.

Mori et al. (2018) approached the problem from a different angle and instead of adjusting virtual content to match the environment, they controlled the opacity of a liquid crystal shutter on an OST-HMD to uniformly dim the real environment so that the HMD content was perceptually brighter.

Hiroi et al. (2017) further employed an automated per-pixel brightness adjustment mechanism in the context of vision augmentation using an OST-HMD. They use an occlusion mask and HMD content for over- and under-exposed regions to make them re-appear in the user view.

However, none of these take user's pupil adaptation into consideration. Several patents related to automated brightness adjustment currently exist (Uhlhorn (2010); Capener (2013); McCulloch et al. (2014)), even going so far as to claim a pupil-based adjustment method, but none of the patents have actually tested the proposed algorithms with a participant group or produced tangible results showing how the eye functions in an AR/MR scenario with variable lighting. Moreover, few of these techniques have actually made it into commercial devices, further demonstrating the need for a formal study and more careful algorithm design.

5.2.3 Further Motivation

Although other vision augmentation and object enhancement methods exist for object enhancement (Hiroi et al. (2017)) or lighting reproduction, this work focuses on user-centric content such as e-mails, world-registered windows such as browsers, and interactive icons or widgets that are more subject to a user's preferred brightness setting.

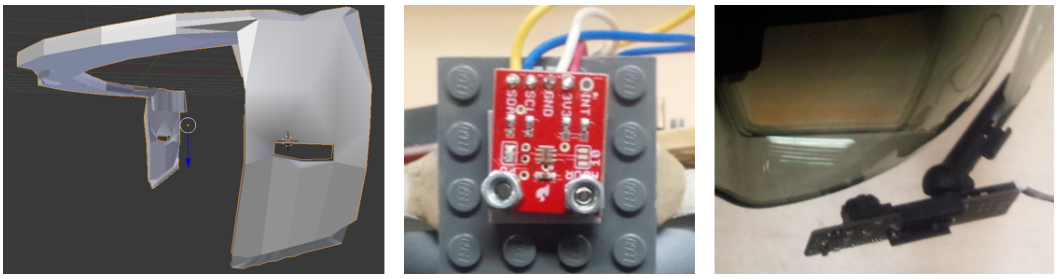


Figure 5.3: Images showing the mount and accessories used in the experiment. Left: The 3D model used to print the mount for the HMD. Center: The TSL2561 light sensor. Right: The Pupil Labs eye tracking camera.

Of the automated methods available, environmental luminance is usually only taken into account. Because the pupil itself is a type of biological pinhole camera, the amount of light passing through to the retina from a particular point in the environment is to a great extent represented by the pupil size. In this work, it is hypothesized that the pupil size is also an essential factor for properly managing the HMD brightness. Accordingly, the system is designed to use the pupil size as an input.

5.3 Hardware and Software Setup

The system is primarily composed of the display, eye tracker, and light sensor for hardware, which are integrated with the eye tracking software, pupillometric light adjustment algorithms, and communication software framework. These parts are described in detail below.

5.3.1 Hardware

Since the method is primarily designed for OST-HMDs, the Microsoft HoloLens is chosen as the test display. To handle interchangeable attachment of both the eye tracking cameras and a variety of forward-facing cameras and light sensors, a custom 3D printed mount is built based on the ModulAR framework (Orlosky et al. (2015)), as shown in Fig. 5.3. This allows the user to attach a variety of sensors or cameras on the same optical axis as virtual content.

To detect the environment light, the front-facing camera of the HoloLens or a typical webcamera is not capable due to the limited dynamic range. Though a high dynamic range (HDR) camera would have worked, its size, weight and cost are not optimal for use with an HMD. Accordingly, a fingernail sized HDR luminosity sensor TSL2561 is chosen for such purpose. It provides a

dynamic range of 0.1 [lx] to 40,000 [lx], and can be sampled every 10 [ms], much better than any commodity web camera could provide. The sensor is controlled with an Arduino Red Board, part number DEV-13975.

For the eye tracking camera, a 60Hz Pupil Labs camera in a single-eye configuration is used, as can be seen in the lower right of Fig. 5.3. Shown on the left of the same figure, the 3D printed mount allows both camera and sensor to be rigidly fixed to the display so that the relative position does not change during use. Both the eye tracking and rendering threads are processed on a desktop personal computer (PC) with an Intel Xeon E5-2690 CPU and an NVIDIA GeForce GTX 680 GPU.

5.3.2 Software

A 3D eye tracking framework developed by Itoh et al. (2016) is used to track the user's eye. The eye tracking provides relatively stable calibration as well as the real size of the pupil in millimeters calculated from a 3D eye model. This eye tracking data is then sent to Unity (5.6.0f3) over a socket on the same PC to minimize latency.

Within Unity, several different types of content is displayed for the experiments such as a browser (billboard) and regular text, as described later in Sec. 5.6. These images are then sent via network socket to the HoloLens using the remote application, which allows content from Unity on the host PC to be directly displayed on the HMD screen. The whole system runs at an average frame rate of 60 frames per second (FPS).

In the experimental setup, all the virtual images are displayed at a distance of 2 [m] directly in front of the user's view point (screen center). Also, since the adjustable brightness range of the HoloLens is subdivided into levels, the brightness level is set to maximum. As such, the perceived brightness of the rendered content is controlled with its opacity adjusted in Unity.

5.4 Pilot Test

Though there already exist much information about how the pupil functions in response to light, a pilot test is conducted for better understanding of how this could be used to manage virtual lighting.

5.4.1 Pupil-based Algorithm

As an initial test, a basic algorithm is built to understand whether pupil size alone could be used to manage lighting and to gain a better understanding of how the pupil would function in dynamic conditions. In essence, the minimum and maximum size of a user's pupil is firstly measured as a baseline, and all luminance values of the display are then linearly interpolated (inversely) between these two values based on the pupil size. For example, the minimum pupil size (full constriction) will result in full display brightness, and maximum pupil size (full dilation) will result in minimum display brightness plus an offset. This offset is included so that complete darkness would not result in completely transparent content, and the offset is fixed at 20% above minimum brightness, selected through initial testing.

5.4.2 Participants, Setup and Conditions

10 subjects, 6 male and 4 female, ranging in age from 22 to 27 (avg. 24.38, stdev. 2.06), participated in the pilot experiment. All subjects had normal or corrected-to-normal vision and stated that they had little to no experience in the usage of OST-HMDs (fewer than 3 uses). The experiment consists of two phases: 1) a comparison of user selected brightness, pupil-based algorithm determined brightness, and mean/max brightness, and 2) subjective ratings of the latter three brightness management strategies to determine how close each is to the ideal brightness setting. The procedure of the experiment is approved by the institutional review board of Osaka University Institutional Review Board.

To display and control real world background lighting, a projector (RICOH PJ WX4141NI) is chosen that can output up to 3,300 [lm] of light. Images are projected with four different brightness levels as shown in (a) through (d) Fig. 5.4, which respectively correspond to the Dark, SlightlyDark, Slightly-Bright, and Bright conditions in Fig. 5.5. To figure out how far off from the ideal the pupil based algorithm is, its results are compared to the mean and maximum brightness settings of the display. In addition to verifying the basic effectiveness of pupil response as a method for lighting adjustment, this could also help with improving the fundamental structure of future iterations of the algorithm and determining what to include in the primary higher dynamic range experiment.

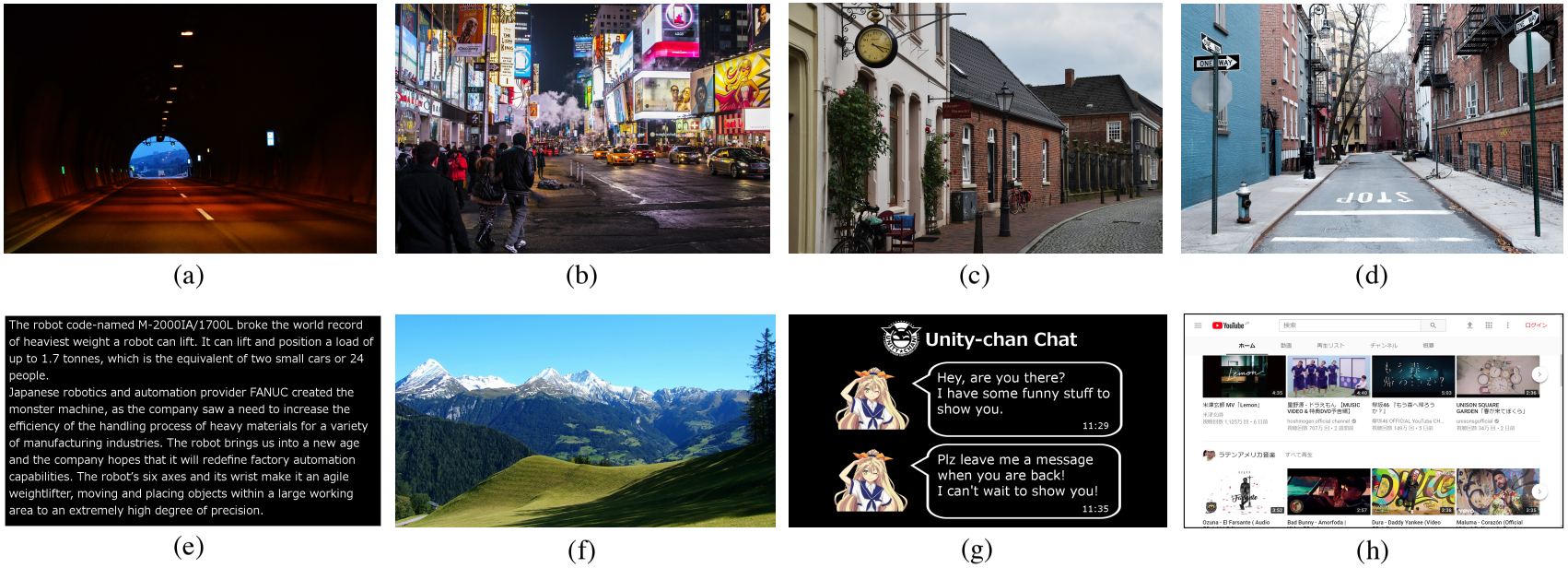


Figure 5.4: The images used in both experiments. (a), (b), (c), and (d) are projected as backgrounds in the pilot experiment, and correspond to the Dark, SlightlyDark, SlightlyBright, and Bright conditions. (e), (f), and (g) are displayed as the overlaid virtual content for the pilot experiment, from left to right: Text Only, Image Only, and Text with Image, which correspond to the result graphs in Fig. 5.5. The final experiment tests (e) and (h). Note that black appears as transparent on the HoloLens.

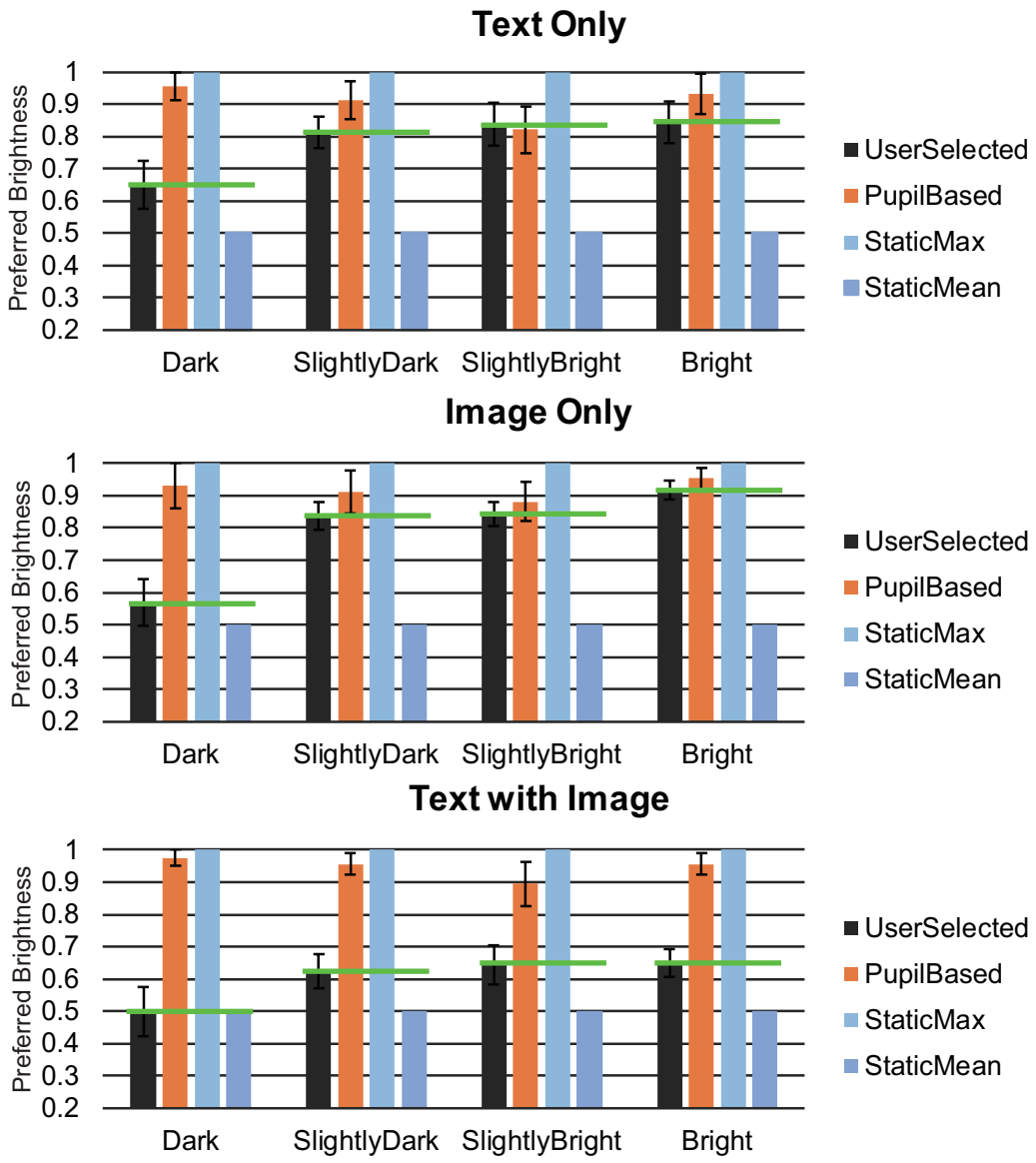


Figure 5.5: Graphs showing how the initial pupil-based algorithm (orange) performs in comparison to mean and max brightness settings. User preferences are taken as ground truth and are shown in black, with extended green lines for reference. The vertical axis represents the minimum (+20% offset) and maximum display brightness of the HoloLens for a given virtual image, scaled from 0 to 1.

5.4.3 Initial Results

Though somewhat simplistic, the evaluation shows that the pupil-based light management is already better than maximizing brightness in some cases.

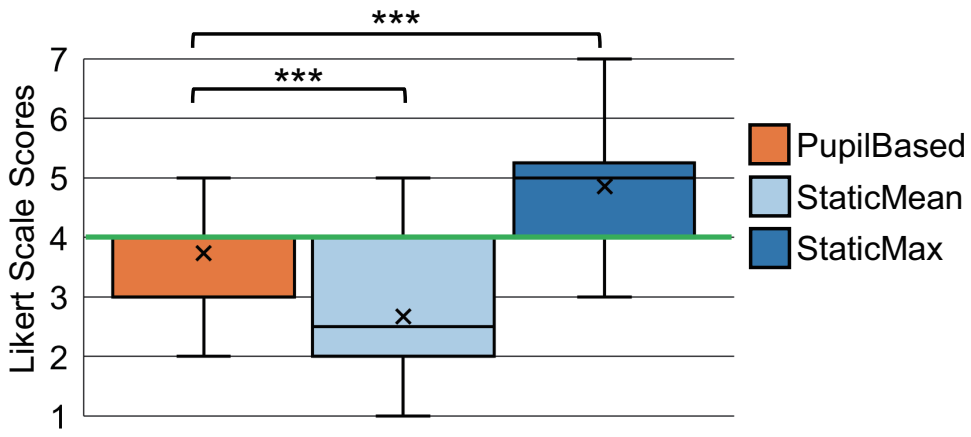


Figure 5.6: Box plot showing subjective results from the second part of the pilot experiment, where participants rated all methods on a seven point Likert scale ranging from too dim (1) to too bright (7). The green line (at 4) represents the "just right" level. The 'x's represent the mean values.

Fig. 5.5 shows a summary of the options most closely matched with user preference, with the best options for each piece of HMD content circled with a dotted green line. The pupil-based algorithm closely (75%) matches user preference for the text only and image only cases, but it is not able to account for low-light situations properly. Fig. 5.6 shows subjective ratings versus the mean and max values for screen brightness. Further refinement and experimentation are necessary, so the algorithms are re-designed and a more thorough experiment is conducted next.

5.5 Algorithm Redesign and Refinement

Prior to finding an algorithm that works well, other implementations were tested but turned out to be ineffective, but a great deal in the process could be learnt.

5.5.1 First Iteration

The first was a hard coded approach that used a weighted function to determine the ideal display brightness. The basic algorithm included a pre-calibrated pupil-brightness model that weighted pupil size and environment light. Through informal testing of this algorithm, it was learnt that the output function was still relatively 1-dimensional and could not handle the entire

spectrum of brightness solely based on the two weighting functions. This was primarily due to two downsides of the approach, including that there was no good way to calibrate the weight function for each user and that the model could not be updated to account for non-linear differences in preference at various brightness levels. Based on these findings, a machine learning approach is explored that matches brightness preference to pupil size and environment lighting over a higher dynamic range.

5.5.2 Machine Learning Approach using Pupil-light Pairs

The high level idea behind this approach is to obtain a mapping of the user's preference for any given pupil size and environment light for a given piece of virtual content. Although many general models exist that can describe the connection between perceived luminance and pupil response (Winn et al. (1994); Watson and Yellott (2012)), it is hard to apply these models directly to the lighting adjustment of OST-HMDs. One reason is that it is difficult to properly measure the luminance of both the near-eye virtual content and the environment in real time. Additionally, the individual differences in users' light adaptation and preferences make it necessary to define a personalized model.

To overcome the limitations of these other approaches, a neural network is implemented to generate a personally optimized model obtained from a short user training phase. This training could be done automatically over the course of a few hours of regular HMD use, but a training period of about 20 minutes (approx. 100 selections over 10 backgrounds x 2 lighting conditions) is defined in the experiment in order to shorten the experimentation time. In the primary experiment, a custom implementation of neural network is used that has 2 hidden layers, limited to 10 neurons for each to reduce the time cost of training and the over-fitting. Back propagation algorithm is used for calculating the weights of the network and the hyperbolic function *tanh* is used as the activation function. All data are trained at a learning rate of 0.003 out of 5000 epochs with the batch size being 1.

The neural network is trained on pupil-light input pairs, i.e., a pupil radius [mm] and environment illuminance [lx] with a user selected preferred brightness (from 0 to 1 in Unity). Through an initial test data set, it is found that passing the raw inputs through a linear scaling transformation without normalizing the distribution works best for tuning the model. Thus, the network can be trained to mimic the user's choices fairly well, and much better than current linear models, as later shown by the experimental results.

5.5.3 Automated Adjustment Filter

A good HMD lighting model needs to be sensitive to all environment conditions, but must also provide a desirable user experience. Because human pupil size is not stable over time due to the adaptation of the cones and rods to incoming light and because the output of the eye tracker is not always perfect, making direct adjustments using the raw pupil value resulted in noisy output and very noticeable changes in HMD lighting. In order to avoid the excessive changes in brightness over short periods of time, a simple filtering (smoothing) algorithm is implemented to account for the variation $\dot{v}$ in pupil size. This is given by the following

$$\dot{v} = \frac{\sum_{\mathrm{T-I}}^{\mathrm{T}} p - \sum_{\mathrm{T-2I}}^{\mathrm{T-I}} p}{\sum_{\mathrm{T-2I}}^{\mathrm{T-I}} p}, \tag{5.1}$$

where $\sum_{\mathrm{M}}^{\mathrm{N}} p$ denotes the summary of the pupil size p from time M to N, T denotes the respective time and I denotes a constant time interval that was picked through trial and error. In the primary experimental setup, the interval I is set to 0.5 [s] and $\dot{v}$ is set to 0.0004 as a result from an initial test. Informal testing of this algorithm appeared to yield much better results, so an in-depth user study for further evaluation is designed as follows.

5.6 User Study

This primary user study is intended to test how the Intellipupil algorithm, referred to as IntelliPupil, would perform in a more complex real environment. For comparison, a linear adjustment model, referred to as Linear, is implemented that sets the display brightness based on the illuminance of the background as measured by the HDR luminosity sensor. The end goals of the user study are to 1) train the IntelliPupil algorithm, 2) verify the efficacy of the trained algorithm and compare the results to the commonly used Linear adjustment, and 3) learn about the behavior of the pupil in the process of OST-HMD usage.

5.6.1 Setup and Participants

The general experiment procedure involves first gathering a training set of data, then evaluating IntelliPupil against the environment light based linear model to see which more closely matches user selections. The layout and

two fisheye views of the experiment environment are shown in Fig. 5.7. A 5x5 meter room is set up where a single window is visible and facing near-direct sunlight to mimic outdoor lighting, as can be seen in the upper right of each image. The other side of the room is darkened to obtain lower lighting conditions. All experiments are conducted from 10 a.m. to 5 p.m. in order to obtain outdoor light. The illuminance value (as taken from the sensor) ranged from approximately 0.002 [lx] to 1000 [lx], which corresponds to point values of 0.5 [cd/m^2] for the darkest corner of the room to 4500 [cd/m^2] when facing the open window, measured with a Konica Minolta CS-100A luminance meter. With this large range of lighting values, both indoor and outdoor lighting could be tested to a certain extent test. The procedure of the experiment is approved by the institutional review board of Osaka University Institutional Review Board.

A total of 12 individuals participated in the experiment, but data from 3 individuals was excluded from analysis due to excessive mascara, irregular pupil shape, and failure of the eye tracker, leaving nine valid participants, 6 male and 3 female ranging in age from 23 to 54 (avg. 28, stdev. 9.33), for analysis.

5.6.2 Task: Preference Selection

The tasks in this experiment are different from the pilot in that participants are required to gaze at physically printed numbers and other objects in the real world rather than a projected background. The number of virtual overlays is also reduced to two, including a browser window (billboard-style) and regular text. This helps to ample training time for InterlliPupil, a period for testing, and subjective evaluation.

5.6.2.1 Linear Model for Comparison

Current linear models either try to match environment light with display brightness as closely as possible or maximize contrast against the environment. Since it was preferred to avoid the problem of preventing the pupil from light-adapting, this work ignored contrast maximization and implemented the model that matches environment light as closely as possible. The algorithm adjusts the display brightness to b_{d} linearly based on the illuminance i of the environment measured by the forward-facing luminosity sensor. This is given by the following

$$b_{\mathrm{d}} = MIN(1, \frac{i}{\mathrm{I_{MAX}} - \mathrm{I_{MIN}}}), \tag{5.2}$$

where $\mathrm{I_{MAX}}$ and $\mathrm{I_{MIN}}$ denote the maximum and minimum illuminance of the environment. In the experiment, $\mathrm{I_{MAX}}$ was set to 0 lx (no light incoming) and $\mathrm{I_{MIN}}$ was set to 1000 lx, a high illuminance level corresponding to the maximum brightness of the HMD.

5.6.2.2 Training

The participants first sat down on a swivel chair, put on the display, adjusted it to the appropriate nose height, and confirmed that virtual content was visible. Next, the eye tracking software was initialized to make sure it was stable. Next, the training phase was started for the experiment, in which participants had to gaze at each of the numbers in the room sequentially. When focused on each number, they used a hand-held mouse to select their preferred brightness for that number by using the scroll wheel to adjust brightness and clicking to make a selection. This click selection was repeated five times per number, and the participant then rotated in the chair so that the HMD faced the next number. This was carried out for times for each number for two general lighting conditions and two HMD conditions as follows:

- Lighting 1: Room lighting on
- Lighting 2: Room lighting off
- HMD 1: Plain text paragraph (non-billboard)
- HMD 2: Browser content (full-screen billboard)

This setup provided 100 (5 x 10 x 2) data points (pupil-light correspondences) for each of the Plain Text and Browser Content conditions. Afterwards, while the participant took a short break, the recorded pupil-light data pairs were fed into the neural network as training data prior to the second half of the experiment.

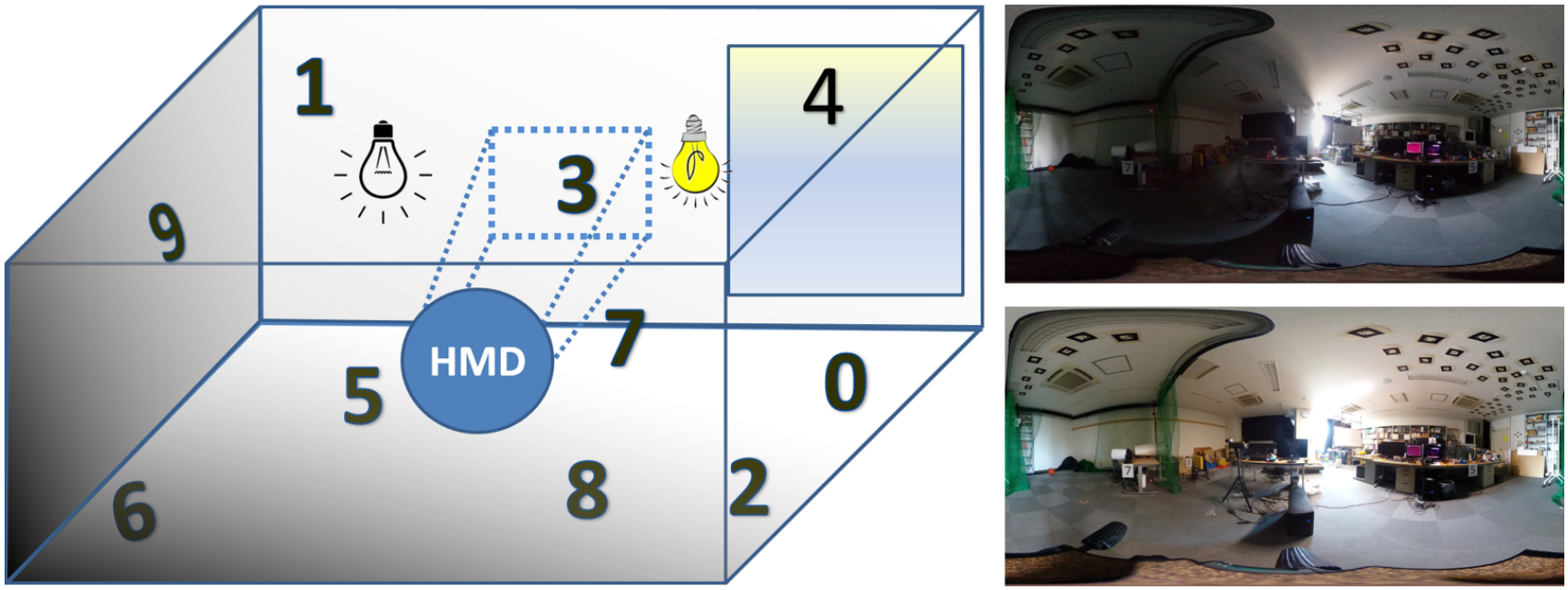


Figure 5.7: Images showing the layout of the experiment and room used to control lighting conditions. Numbers printed on single sheets of paper were disbursed throughout the room so that all participants would have a high dynamic range of lighting conditions (approximate range of $0.5\,\mathrm{cd/m^2}$ in the corner to $4500\,\mathrm{cd/m^2}$ in the window). Lights-off (up-right) and lights-on (bottom-right) conditions are also shown, taken with a 360° camera.

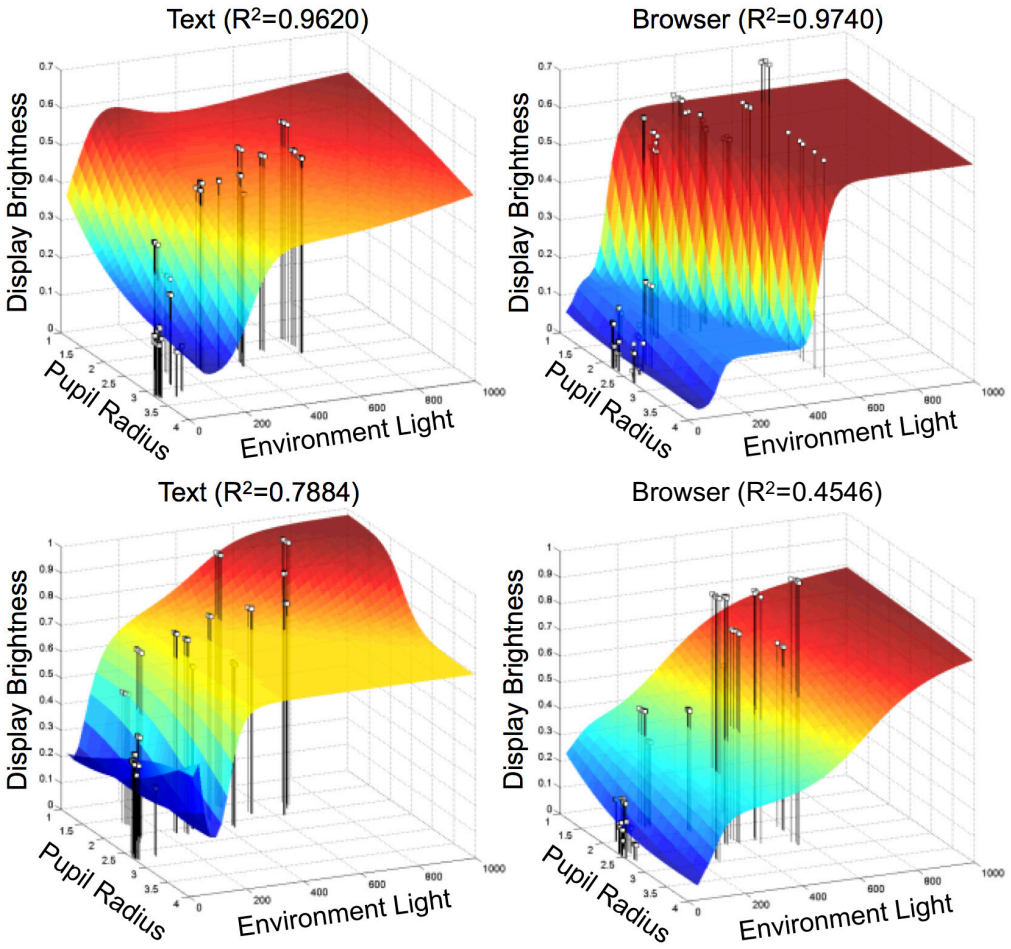


Figure 5.8: 3D visualizations of the output from some of the best and worst fits for the trained neural network model, including all possible pupil-light input pairs. The white elevated points are the user input pairs that trained the model. Pupil radius is in [mm], environment light is in [lx], and HMD brightness is the resulting alpha value from 0-1 in Unity.

5.6.2.3 Algorithm Evaluation and Comparison

The second part of the experiment was designed to determine how well the IntelliPupil algorithm could actually reproduce user preferences and to see how this compared to the linear model. This was designed as more of an "in the wild" evaluation, where participants were asked to gaze at multiple objects/directions for both the Linear and IntelliPupil approaches. This time, instead of choosing a preference, participants were asked to rate how close the algorithmic result was to their actual preference, with 4 being perfect, 1 being

too dim, and 7 being too bright, using the mouse wheel to select a score. Both algorithms were displayed at random, one after the other for a particular gaze point, so that participants could rate each of for each pair over the same background. This was repeated 20 times and both types of HMD content in an intermediate lighting condition (half of the room lights on) to expose participants to a lighting condition that was not part of the training data set.

5.6.3 Results

To observe how well the neural network fit users' initial selections, the average accuracy of all pupil-light data pairs for each participant was estimated by calculating the mean squared error (MSE) and coefficient of determination (R^2) of Plain Text and Browser Content. Table 5.1 shows the average results aggregated from all rounds of training in the primary experiment, which resulted in relatively high correlations between user selections and model output. Correlations were relatively good for most users, and are visualized for some of the best and worst cases in Fig. 5.8, with input pairs as white elevated points and the resulting neural network outputs (for all possible inputs) as surface plots.

5.6.3.1 Comparison of IntelliPupil and Linear Model

Subjective ratings for the Linear and IntelliPupil models are summarized in Fig. 5.9, where a score of 4 represents the ideal (just right) brightness. A Kruskal-wallis test for subjective ratings showed a significant difference between Linear and IntelliPupil for both the browser ($\chi^2 = 94.211 > \chi^2_u = 6.635, p < 0.01$) and text ($\chi^2 = 141.337 > \chi^2_u = 6.635, p < 0.01$) condi-

Table 5.1: Neural network training result for each participant. Note that the output preferred brightness ranges from 0 to 1.

	Plain Text		Browser Content	
#	MSE	R^2	MSE	R^2
1	0.0045	0.8471	0.0048	0.7363
2	0.0068	0.8844	0.0093	0.7294
3	0.0111	0.7957	0.0046	0.9510
4	0.0007	0.9000	0.6505	0.4546
5	0.0113	0.7884	0.0057	0.9131
6	0.0082	0.9134	0.0016	0.9740
7	0.0147	0.8847	0.0036	0.9148
8	0.0011	0.9620	0.0209	0.7077
9	0.0035	0.9055	0.0012	0.9193

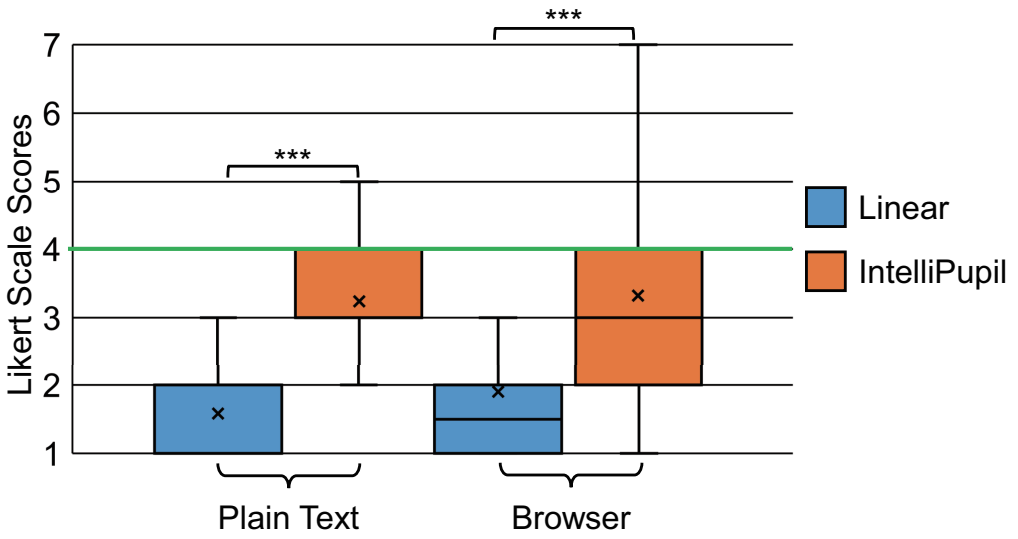


Figure 5.9: Box plots of average brightness selections relative to ideal (rating of 4) for Linear and IntelliPupil methods and browser versus text.

tions. This evidence shows that IntelliPupil outperformed the linear model at matching user preferences, though ratings were still slightly below ideal. It was also realized after the fact that the hard-coded scaling parameter for the linear model may have been too strict since I_{MAX} was set to 1000 lx but observed maximum illuminance from the experiment data was around 800 lx. This means that the Linear adjustment likely only provided approximately 80% of its maximum output, which lead to the lower subjective scores.

5.7 Discussion

5.7.1 Implication

Based on the visualizations in Fig. 5.8, it is inferred that 1) the brightness preferences of users for various environment lighting are non-linear, and 2) though somewhat similar, models for individual users can vary greatly. This has significant implications for the way content designers should choose to display text or other content, especially since increasing contrast or matching background luminance is currently a common approach in both research and industry.

5.7.2 Limitation and Future Work

From the results in Fig. 5.9, it is inferred that the IntelliPupil output still deviates somewhat from the user's ideal display brightness. Several possibilities remain for this inaccuracy, including the need for additional training of the model in real world use, user emotional state, and other unknown parameters that may need to be included in the model. One example of such a parameter might be the number of lit pixels on the screen. Since the tested content were relatively static blocks of content (text/browser), further development is required for a function that takes each pixel into account and would thus not need to trained separate data sets for different virtual content. Display brightness as well as FoV will also have a large influence on pupil size. Large bright windows displayed in a 100 degree FoV display in a dark environment will likely be more influential on pupil size than a small icon overlaid onto a scene in broad daylight. Future work should strive toward a content-independent model that can be trained using more flexible inputs, though this will need to be heavily optimized.

In the primary experiment, a natural scene was used as the background and asked the participants to gaze in various directions while evaluating the algorithms to get practical results. However, it is also believed that doing the experiment in a more controlled condition could provide deeper insights into how the lighting conditions of the environment and HMD would affect pupil size, which is planned as future work. It is necessary to conduect a further user experiment under fully controlled and more various lighting conditions to reveal the relationship between the environment/HMD brightness and the pupil adaptation when one's using an OST-HMD outdoors.

Lastly, though a 20-minute training session was designated for users during the experiments, it was available to just as easily use brightness adjustments over time to train the algorithm. In other words, the adjustments and button presses have to do to adjust display lighting anyway could be fed into the algorithm without the user noticing, eliminating a formal training phase. For practical use, it is necessary to come up with a more stable and drift-independent eye model. It is believed that some of the results with poor fitting, such as subject #4 with the browser content in Table 5.1, which is also visualized as the lower right of Fig. 5.8, were due to a re-calibration caused by a large positional shift of the eye tracking camera during the experiment. Instead of initializing the tracker for each user prior to or during training, it would be ideal to save a user's previously generated eye models for longer term use. In the future, it is also a goal to refine the algorithm so it can be applied to previously unseen content and for users without any

training information.

5.8 Chapter Conclusion

This work presents IntelliPupil, an algorithm that can automatically adjust display lighting a user's brightness and contrast preferences. Using both pupil size and environment light, IntelliPupil can adjust HMD lighting better than current linear models, which shows that user preferences for contrast must be taken into account in addition to environment light. Experiment results also showed that environment light alone is not enough to manage display lighting, and that pupil size is an equally important factor in adjusting to an ideal brightness. It is expected that the IntelliPupil algorithm, the neural network design, and the experiment results will change the way people think about light management techniques for OST-HMDs in the near future.

Chapter 6
Conclusion

This dissertation presents a theory of developing and applying eye-based user interfaces to HMDs for hands-free usability and improved user experience. This theory postulates that automated methods for light modulation and pure gaze-based manipulations can outperform traditional methods for HMD interaction. Two methods are deployed for the attempt, including: 1) a method that makes use of pupil response for automated light modulation of OST-HMDs, and 2) a method that enables pure gaze-based manipulation of 3 DoF. A follow-up study also explores the usability of different eye gaze-based object rotation methods that are suitable for HMDs.

6.1 Summary

A summary of the contributions and findings made by this work is shown as follows.

Gaze-based Three-dimensional Object Manipulation. This work proposes a novel method called OrthoGaze that allows the user to intuitively manipulate the three-dimensional position of a virtual object using only the eye or head gaze. This approach makes use of three selectable, orthogonal planes, where each plane not only helps guide the user's gaze in an arbitrary virtual space, but also allows for 2-DoF manipulations of object position. This work conducted two user studies involving aiming and docking tasks in VR to evaluate the fundamental characteristics of sustained gaze aiming and to determine which type of gaze-based control performs best when combined with OrthoGaze. In summary, the main contributions of this work are:

- This work presents a novel approach that enables hands-free adjustment of virtual object position of 6 DoF in HMDs.
- An experiment is conducted that evaluates sustained eye-gaze and head-gaze aiming in a painting task. The results show that eye-gaze outperforms head-gaze in terms of accuracy. Furthermore, in some cases larger areas can be covered with eye-gaze than head-gaze.

- The experimental results show that for 3D docking tasks, eye and head gaze-based control with OrthoGaze can achieve 78% and 96% success rates, respectively, when compared to a hand-held controller.

Gaze-based Three-dimensional Object Rotation. This work explored three methods to handle rotations of virtual objects using gaze, including RotBar: a method that maps line-of-sight eye gaze onto per-axis rotations, RotPlane: a method that makes use of orthogonal planes to achieve per-axis angular rotations, and RotBall: a method that combines a traditional arcball with an external ring to handle user-perspective roll manipulations. This work validated the efficiency of each method by conducting a user study involving a series of orientation tasks along different axes with each method. In summary, the contributions of this work are:

- Three different methods are implemented that improve upon existing work and redesigned them to specifically address the needs of those who need eye-only control to rotate virtual objects.
- A user study is conducted that compares these methods in different situations to determine their suitability for simple and complex object manipulations. The results show that RotBar and RotPlane are more suitable for simple rotations and RotBall is more suitable for complicated manipulations.
- Results of the user study revealed several observations that need to be considered by future interfaces for gaze-based manipulation.

Automated Light Modulation for OST-HMDs. In this work, a novel algorithm called IntelliPupil is proposed to properly modulate augmentation lighting for a variety of lighting conditions and real scenes. The system first takes data composed of real scene luminance and changes in the pupil diameter as passive inputs. The data is coupled with user-controlled brightness selections, allowing the algorithm to generate a model fitting to the user preference using a feed-forward neural network. Using a small amount of training data, both the scene luminance and the pupil size are used as inputs into the neural network, which can then automatically adjust to a user's personal brightness preference in real time. In summary, IntelliPupil involves the following advantages to light modulation of OST-HMDs.

Dynamic to Use Cases. IntelliPupil detects the changes in the environment brightness according to the input pairs of scene luminance and pupil

response, and then adjusts the display automatically for every frame. As a result, the screen is kept to a optimal brightness level dynamic to the use case.

Hands-free. The whole IntelliPupil system is usable in a completely hands-free way. The auto-adjustment takes passive inputs from the luminance sensor and the eye tracking camera, which requires no manual effort at all. As for the initial calibration, the history of user selections for the preferred brightness can also be recorded through hands-free methods, e.g. a gaze-based bias selection interface that provides the up/down adjustment of the display brightness.

Better Matching with User Preference. The experimental results show that IntelliPupil significantly outperforms a linear adjustment method in matching the HMD brightness with a user-preferred level. This is essential for practical use cases of OST-HMDs.

The main contributions of this work include:

- This work proposes a novel algorithm that accounts for user preference, pupil size, and environment light to automatically manage display brightness of OST-HMDs.
- The experimental results reveal that pupil size is just as important as environment light for optimizing brightness and that IntelliPupil outperforms linear adjustment methods in matching the user preference of the display brightness.

In general, this work gives some implications regarding applying eye tracking for achieving hands-free HMDs. It shows that methods following certain designs can address the need for eye gaze-based object manipulations of high DoF. With the experimental results optimistic, the author believe that it is viable to develop a eye-based 3D CAD system on HMDs that would benefit from the immersive experience of HMDs along with the hands-free modality of eye-based control. In addition, results from Chapter 5 reveal that eye-based human factors can serve as an additional cue for automatically tuning an HMD (brightness in the case of this work) to improve its functionality. It is worthwhile to dig deeper into this unexplored topic for more novel insights that could help evolve the next-generation human-HMD interaction.

While eye tracking is becoming more viable for HMDs, the author hopes that the findings of this work can contribute to eye-based infrastructure in future HMDs.

6.2 Suggestions for Future Work

This work with its findings has taken a step forward towards achieving hands-free usability of HMDs. However, there are still various tasks left regarding eye-based interaction for HMDs. This section outlines two concrete ideas accordingly for inspiring further research in the same domain.

Eye-based Context-aware Augmentation. One of the main purposes of AR is to provide valuable information in real-time to the user, which asks for the augmentation being proper, flexible and dynamic, say "intelligent and attentive enough", to arbitrary situations. For example an ideal AR guidance system should be able to read the user's activities (attention, interests and needs) and provide proper and adequate guidance information in real-time. To achieve such in-situ AR requires accurate and comprehensive tracking of various elements in real-time, such as the surroundings, users' tasks and interests and so on.

As such, eye tracking is expected to be valuable and applicable for such goals, since tracking eye movements can provide us with key information for interpreting and understanding a person's interests, needs, cognitive states and affective states in real-time. Besides, eye tracking also allows for the analysis of relation between the environment and eyes, which can play an important role in further understanding of human behaviours (Nishino and Nayar (2006); Nitschke et al. (2013)). In this work, Chapter 5 made one attempt towards such intelligent context-aware AR systems, though there still remain valuable tasks unresolved or undiscovered in the same domain. The author believes that relating the eye tracking technology could help with the development of context-aware AR systems and lead to better user experience of HMDs.

Unified Hands-free Interaction Frameworks. Recently, more and more industries are promoting HMDs with eye tracking cameras integrated, e.g. HTC Vive Pro Eye and Microsoft HoloLens 2. It is promising that eye tracking is capable of becoming a basic means of interaction with HMDs in future. However while various studies, including this work, is promoting novel eye-based interaction methods in drops, there still lacks an unified framework allowing for comprehensive hands-free interaction, for example a framework that allows the user to use a combination of eye/head movements and voice to accomplish all available interaction commands while using the HMD.

To reach this goal, large-scaled development, prototyping, and a large

number of task-based/user-based tests are necessary, which requires long-term efforts. However, it can be predicted that the unified hands-free interaction can contribute to the evolution of future human-HMD interaction to a large extent.

Appendix A

Subjective Surveys

A.1 OrthoGaze User Survey

About Participant

Gender
◯ Male
◯ Female

Are you short-sighted?
◯ Yes.
◯ No.

Are you wearing vision correction glasses or contact lenses?
◯ Yes.
◯ No.

Do you have any (partial) color blindness?
◯ Yes.
◯ No.
◯ I'm not sure.

How is your experience on using head-mounted displays before this experiment?
◯ Never used before.
◯ Less than 5 times.
◯ 5 to 10 times.
◯ More than 10 times.

How is your experience on eye gaze tracking before this experiment?
◯ Never used before.
◯ Less than 5 times.
◯ 5 to 10 times.
◯ More than 10 times.

Painting Task

About Task

The painting task was easy.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

"Eye Gaze Mode"

The "Eye Gaze Mode" helped me solve the painting task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Eye Gaze Mode" was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Eye Gaze Mode" was of low fatigue.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

"Head Gaze Mode"

The "Head Gaze Mode" helped me solve the painting task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Head Gaze Mode" was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Head Gaze Mode" was of low fatigue.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

Docking Task

About Task

The docking task was easy.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The orthogonal planes helped me solve the docking task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The orthogonal planes were visually not distracting.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

"Eye Gaze Mode"

The "Eye Gaze Mode" helped me solve the docking task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Eye Gaze Mode" was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Eye Gaze Mode" was of low fatigue.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

"Head Gaze Mode"

The "Head Gaze Mode" helped me solve the docking task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Head Gaze Mode" was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Head Gaze Mode" was of low fatigue.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

"Controller Mode"

The "Controller Mode" helped me solve the docking task.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Controller Mode" was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

The "Controller Mode" was of low fatigue.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ ◯ ◯ Strongly agree.

Please write down here if you have any comment regarding the experiment.

_____

_____

_____

_____

A.2 Gaze-based Rotation User Survey

About Participant

Gender
◯ Male
◯ Female

Are you short-sighted?
◯ Yes.
◯ No.

Are you wearing vision correction glasses or contact lenses?
◯ Yes.
◯ No.

Do you have any (partial) color blindness?
◯ Yes.
◯ No.
◯ I'm not sure.

How is your experience on using head-mounted displays before this experiment?
◯ Never used before.
◯ Less than 5 times.
◯ 5 to 10 times.
◯ More than 10 times.

How is your experience on eye gaze tracking before this experiment?
◯ Never used before.
◯ Less than 5 times.
◯ 5 to 10 times.
◯ More than 10 times.

How do you think of your mental rotation skill?
Extremely bad. ◯ ◯ ◯ ◯ ◯ Extremely good.

RotBar

I think that I would like to use this system frequently.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system unnecessarily complex.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought the system was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I think I would need the support of a technical person to be able to use this system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the various functions in this system were well integrated.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought there was too much inconsistency in this system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I would imagine that most people would learn to use this system very quickly.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system very cumbersome to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I felt very confident using the system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I needed to learn a lot of things before I could get going with this system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

RotPlane

I think that I would like to use this system frequently.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system unnecessarily complex.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought the system was easy to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I think I would need the support of a technical person to be able to use this system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the various functions in this system were well integrated.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought there was too much inconsistency in this system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I would imagine that most people would learn to use this system very quickly.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system very cumbersome to use.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I felt very confident using the system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I needed to learn a lot of things before I could get going with this system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

RotBall

I think that I would like to use this system frequently.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system unnecessarily complex.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought the system was easy to use.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I think I would need the support of a technical person to be able to use this system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the various functions in this system were well integrated.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I thought there was too much inconsistency in this system.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I would imagine that most people would learn to use this system very quickly.

Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I found the system very cumbersome to use.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I felt very confident using the system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

I needed to learn a lot of things before I could get going with this system.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

About Experiment

The task was easy.
Strongly disagree. ◯ ◯ ◯ ◯ ◯ Strongly agree.

In general, how do you prefer (rank) the three systems?.
RotBar ___
RotPlane ___
RotBall ___

Please write down here if you have any comment regarding the experiment.

_____
_____
_____
_____

Bibliography

Alallah, F., Neshati, A., Sakamoto, Y., Hasan, K., Lank, E., Bunt, A., and Irani, P. (2018). Performer vs. Observer: Whose Comfort Level Should We Consider when Examining the Social Acceptability of Input Modalities for Head-Worn Display? In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 10:1–10:9.

Anders, G. (2001). Pilot's attention allocation during approach and landing – eye- and head-tracking research in an a330 full flight simulator. In *Proceedings of the 11th International Symposium on Aviation Psychology*, Columbus, OH, USA.

Argelaguet, F., Hoyet, L., Trico, M., and Lecuyer, A. (2016). The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *2016 IEEE Virtual Reality (VR)*, pages 3–10.

Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123.

Beach, G., Cohen, C. J., Braun, J., and Moody, G. (1998). Eye Tracker System for Use with Head Mounted Displays. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4348–4352.

Bellini, H., Chen, W., Sugiyama, M., Shin, M., Alam, S., and Takayama, D. (2016). *Profiles in Innovation: Virtual & Augmented Reality*. Goldman Sachs Research.

Blattgerste, J., Renner, P., and Pfeiffer, T. (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*, pages 1:1–1:9.

Boff, K. R., Kaufman, L., and Thomas, J. P. E. (1986). *Handbook of Perception and Human Performance, Vol. 2: Cognitive Processes and Performance*. John Wiley & Sons.

Bowman, D., Kruijff, E., LaViola Jr, J. J., and Poupyrev, I. P. (2004). *3D User Interfaces: Theory and Practice*. Addison-Wesley.

Bradley, M. M., Miccoli, L., Escrig, M. A., and Lang, P. J. (2008). The Pupil as a Measure of Emotional Arousal and Autonomic Activation. *Psychophysiology*, 45(4):602–607.

Brooke, J. (1996). Sus-a quick and dirty usability scale. In *Usability evaluation in industry*, pages 189–194. London: Taylor and Francis.

Capener, C. L. (2013). Techniques for Adaptive Brightness Control of a Display. US Patent 8,537,174.

Castle, R. O. and Murray, D. W. (2009). Object Recognition and Localization while Tracking and Mapping. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 179–180.

Chae, H. J., Hwang, J.-i., and Seo, J. (2018). Wall-based space manipulation technique for efficient placement of distant objects in augmented reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 45–52.

Chen, M., Mountford, S. J., and Sellen, A. (1988). A study in interactive 3-d rotation using 2-d control devices. *SIGGRAPH Computer Graphics*, 22(4):121–129.

Colombo, C. and Del Bimbo, A. (1997). Interacting Through Eyes. *Robotics and Autonomous Systems*, 19(3-4):359–368.

De Groot, S. and Gebhard, J. (1952). Pupil Size as Determined by Adapting Luminance. *Journal of the Optical Society of America*, 42(7):492–495.

Drewes, H. and Schmidt, A. (2007). Interacting with the computer using gaze gestures. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II*, pages 475–488.

Duchowski, A. T. (2002). A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments, & Computers*, 34(4):455–470.

Duffy, V. G. and Chan, A. H. (2002). Effects of Virtual Lighting on Visual Performance and Eye Fatigue. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 12(2):193–209.

Fuhl, W., Tonsen, M., Bulling, A., and Kasneci, E. (2016). Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art. *Machine Vision and Applications*, 27(8):1275–1288.

Fujii, K., Grossberg, M. D., and Nayar, S. K. (2005). A Projector-Camera System with Real-time Photometric Adaptation for Dynamic Environments. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 814–821.

Gabbard, J. L., Swan, J. E., Hix, D., Kim, S.-J., and Fitch, G. (2007). Active text drawing styles for outdoor augmented reality: A user-based study and design implications. In *Virtual Reality Conference, 2007. VR'07. IEEE*, pages 35–42. IEEE.

Graeber, D. A. and Andre, A. D. (1999). Assessing visual attention of pilots while using electronic moving maps for taxiing. In *Proceedings of the Tenth Symposium on Aviation Psychology*, pages 791–796, Columbus, OH, USA.

Guestrin, E. D. and Eizenman, M. (2006). General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on biomedical engineering*, 53(6):1124–1133.

Hiroi, Y., Itoh, Y., Hamasaki, T., and Sugimoto, M. (2017). AdaptiVisor: Assisting Eye Adaptation via Occlusive Optical See-through Head-mounted Displays. In *Proceedings of the Augmented Human International Conference*, pages 9:1–9:9.

Ho, G., Scialfa, C., Caird, J., and Graw, T. (2001). Visual search for traffic signs: The effects of clutter, luminance, and aging. *Human factors*, 43:194–207.

Hornof, A., Cavender, A., and Hoselton, R. (2004). EyeDraw: A System for Drawing Pictures with Eye Movements. In *Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility*, pages 86–93.

Huey, E. B. (1908). *The Psychology and Pedagogy of Reading*. The MacMillan Company.

Istance, H., Bates, R., Hyrskykari, A., and Vickers, S. (2008). Snap Clutch, a Moded Approach to Solving the Midas Touch Problem. In *Proceedings of the Symposium on Eye Tracking Research & Applications*, pages 221–228. ACM.

Itoh, Y. and Klinker, G. (2014). Interaction-Free Calibration for Optical See-Through Head-Mounted Displays based on 3D Eye Localization. In *In Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 75–82.

Itoh, Y., Orlosky, J., and Swirski, L. (2016). 3D-Eye-Tracker Source. https://github.com/YutaItoh/3D-Eye-Tracker. Accessed March 12th, 2018.

Jacob, R. J. K. (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*, pages 11–18.

Jacob, R. J. K. (1995). Eye Tracking in Advanced Interface Design. In *Virtual Environments and Advanced Interface Design*, pages 258–288. Oxford University Press, Inc.

Jalaliniya, S., Mardanbegi, D., and Pederson, T. (2015). MAGIC Pointing for Eyewear Computers. In *Proceedings of the ACM International Symposium on Wearable Computers*, pages 155–158.

Johansson, R. S., Westling, G., Bäckström, A., and Flanagan, J. R. (2001). Eye–Hand Coordination in Object Manipulation. *Journal of Neuroscience*, 21(17):6917–6932.

Karl, D., Farhi, M., Krohn, D. P., Schneiderman, J., Soderquist, K., Grant, A., Murphy, B., and Straughan, B. (2019). *2019 Augmented and Virtual Reality Survey Report*. Perkins Coie LLP.

Kassner, M., Patera, W., and Bulling, A. (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*, pages 1151–1160.

Katzakis, N., Seki, K., Kiyokawa, K., and Takemura, H. (2013). Mesh-grab and arcball-3d: Ray-based 6-dof object manipulation. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, pages 129–136.

Katzakis, N., Teather, R. J., Kiyokawa, K., and Takemura, H. (2015). INSPECT: Extending Plane-Casting for 6-DOF Control. *Human-centric Computing and Information Sciences*, 5(1):22.

Khamis, M., Oechsner, C., Alt, F., and Bulling, A. (2018). Vrpursuits: Interaction in virtual reality using smooth pursuit eye movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, pages 18:1–18:8.

Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., and Torralba, A. (2016). Eye tracking for everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2176–2184.

Ktena, S. I., Abbott, W., and Faisal, A. A. (2015). A virtual reality platform for safe evaluation and training of natural gaze-based wheelchair driving. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 236–239. IEEE.

Kumar, M., Winograd, T., Winograd, T., and Paepcke, A. (2007). Gaze-Enhanced Scrolling Techniques. In *Extended Abstracts on Human Factors in Computing Systems*, pages 2531–2536.

Kuo, C., Fanton, M., Wu, L., and Camarillo, D. (2018). Spinal constraint modulates head instantaneous center of rotation and dictates head angular motion. *Journal of Biomechanics*, 76:220–228.

Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., and Billinghurst, M. (2018). Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 81:1–81:14.

Langton, S. R., Watt, R. J., and Bruce, V. (2000). Do the Eyes Have it? Cues to the Direction of Social Attention. *Trends in Cognitive Sciences*, 4(2):50–59.

LaViola Jr, J. J., Kruijff, E., McMahan, R. P., Bowman, D., and Poupyrev, I. P. (2017). *3D user interfaces: theory and practice, 2nd ed.* Addison-Wesley Professional.

Lee, J. W., Cho, C. W., Shin, K. Y., Lee, E. C., and Park, K. R. (2012). 3D Gaze Tracking Method Using Purkinje Images on Eye Optical Model and Pupil. *Optics and Lasers in Engineering*, 50(5):736–751.

Lee, J. Y., Park, H. M., Lee, S. H., Shin, S. H., Kim, T. E., and Choi, J. S. (2014). Design and Implementation of an Augmented Reality System Using Gaze Interaction. *Multimedia Tools and Applications*, 68(2):265–280.

Lee, T. and Hollerer, T. (2007). Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 83–90.

Lee, Y., Shin, C., Plopski, A., Itoh, Y., Piumsomboon, T., Dey, A., Lee, G., Kim, S., and Billinghurst, M. (2017). Estimating Gaze Depth Using Multi-Layer Perceptron. In *Proceedings of the International Symposium on Ubiquitous Virtual Reality*, pages 26–29.

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

Liu, A. (1998). Chapter 20 - what the driver's eye tells the car's brain. In Underwood, G., editor, *Eye Guidance in Reading and Scene Perception*, pages 431–452. Elsevier Science Ltd, Amsterdam.

Liu, C., Plopski, A., and Orlosky, J. (2020). Orthogaze: Gaze-based three-dimensional object manipulation using orthogonal planes. *Computers & Graphics*, 89:1–10.

Majaranta, P. and Räihä, K.-J. (2007). Text Entry by Gaze: Utilizing Eye-Tracking. *Text Entry Systems: Mobility, Accessibility, Universality*, pages 175–187.

Mardanbegi, D., Langlotz, T., and Gellersen, H. (2019). Resolving Target Ambiguity in 3D Gaze Interaction through VOR Depth Estimation. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, page 612.

Mattusch, T., Mirzamohammad, M., Khamis, M., Bulling, A., and Alt, F. (2018). Hidden pursuits: Evaluating gaze-selection via pursuits when the stimuli's trajectory is partially hidden. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 27:1–27:5.

McCulloch, D. J., Hastings, R. L., Geisner, K. A., Crocco, R. L., Balan, A. O., Knee, D. L., Scavezze, M. J., Latta, S. G., and Mount, B. J. (2014). See-Through Display Brightness Control. US Patent 8,752,963.

McNamara, A., Boyd, K., Oh, D., Sharpe, R., and Suther, A. (2018). Using Eye Tracking to Improve Information Retrieval in Virtual Reality. In *Adjunct Proceedings of the IEEE International Symposium on Mixed and Augmented Reality Adjunct*, pages 242–243.

Mine, M. R. (1996). Working in a virtual world: Interaction techniques used in the chapel hill immersive modeling program. Technical report, USA.

Mine, M. R., Brooks, F. P., and Sequin, C. H. (1997). Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pages 19–26.

Mohan, P., Goh, W. B., Fu, C., and Yeung, S. (2018). Dualgaze: Addressing the midas touch problem in gaze mediated vr interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 79–84.

Mohr, P., Tatzgern, M., Langlotz, T., Lang, A., Schmalstieg, D., and Kalkofen, D. (2019). TrackCap: Enabling Smartphones for 3D Interaction on Mobile Head-Mounted Displays. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 585:1–585:11.

Mori, S., Ikeda, S., Plopski, A., and Sandor, C. (2018). BrightView: Increasing Perceived Brightness of Optical See-Through Head-Mounted Displays through Unnoticeable Incident Light Reduction. In *In Proceedings of the IEEE Conference on Virtual Reality*, pages 1–8.

Nakazawa, A. and Nitschke, C. (2012). Point of gaze estimation through corneal surface reflection in an active illumination environment. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., editors, *Computer Vision – ECCV 2012*, pages 159–172. Springer Berlin Heidelberg.

Nanayakkara, S., Shilkrot, R., Yeo, K. P., and Maes, P. (2013). EyeRing: A Finger-Worn Input Device for Seamless Interactions with our surroundingS. In *Proceedings of the Augmented Human International Conference*, pages 13–20.

Nishino, K. and Nayar, S. K. (2006). Corneal imaging system: Environment from eyes. *International Journal of Computer Vision*, 70(1):23–40.

Nitschke, C., Nakazawa, A., and Takemura, H. (2013). Corneal imaging revisited: An overview of corneal reflection analysis and applications. *IPSJ Transactions on Computer Vision and Applications*, 5:1–18.

Orlosky, J., Kiyokawa, K., and Takemura, H. (2013). Dynamic Text Management for See-Through Wearable and Heads-Up Display Systems. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 363–370.

Orlosky, J., Toyama, T., Kiyokawa, K., and Sonntag, D. (2015). ModulAR: Eye-Controlled Vision Augmentations for Head Mounted Displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1259–1268.

Oshima, K., Moser, K. R., Rompapas, D. C., Swan, J. E., Ikeda, S., Yamamoto, G., Taketomi, T., Sandor, C., and Kato, H. (2016). Improved Clarity of Defocussed Content on Optical See-Through Head-Mounted Displays. In *Proceedings of the IEEE Symposium on 3D User Interfaces*, pages 173–181.

Pathmanathan, N., Becher, M., Rodrigues, N., Reina, G., Ertl, T., Weiskopf, D., and Sedlmair, M. (2020). Eye vs. head: Comparing gaze methods for interaction in augmented reality. In *ACM Symposium on Eye Tracking Research and Applications*, ETRA '20 Short Papers, New York, NY, USA. Association for Computing Machinery.

Pfeuffer, K., Mayer, B., Mardanbegi, D., and Gellersen, H. (2017). Gaze + Pinch Interaction in Virtual Reality. In *Proceedings of the Symposium on Spatial User Interaction*, pages 99–108.

Pfleging, B., Fekety, D. K., Schmidt, A., and Kun, A. L. (2016). A Model Relating Pupil Diameter to Mental Workload and Lighting Conditions. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 5776–5788.

Piekarski, W. and Thomas, B. (2004). Augmented reality working planes: A foundation for action and construction at a distance. In *ISMAR 2004: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 162–171.

Piumsomboon, T., Lee, G., Lindeman, R. W., and Billinghurst, M. (2017). Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 36–39.

Plopski, A., Nitschke, C., Kiyokawa, K., Schmalstieg, D., and Takemura, H. (2015). Hybrid eye tracking: Combining iris contour and corneal imaging. In *In Proceedings of the International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, pages 183–190, Kyoto, Japan.

Plopski, A., Orlosky, J., Itoh, Y., Nitschke, C., Kiyokawa, K., and Klinker, G. (2016). Automated Spatial Calibration of HMD Systems with Uncon-

strained Eye-cameras. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 94–99.

Pouke, M., Karhu, A., Hickey, S., and Arhippainen, L. (2012). Gaze Tracking and Non-touch Gesture Based Interaction Method for Mobile 3D Virtual Spaces. In *Proceedings of the Australian Computer-Human Interaction Conference*, pages 505–512.

Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. (1996). The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 79–80.

Poupyrev, I., Weghorst, S., and Fels, S. (2000). Non-isomorphic 3d rotational techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 540–547, New York, NY, USA. Association for Computing Machinery.

Qian, Y. Y. and Teather, R. J. (2017). The eyes don't have it: An empirical comparison of head-based and eye-based selection in virtual reality. In *Proceedings of the Symposium on Spatial User Interaction*, pages 91–98.

Räihä, K.-J. and Špakov, O. (2009). Disambiguating ninja cursors with eye gaze. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1411–1414.

Rompapas, D. C., Rovira, A., Plopski, A., Sandor, C., Taketomi, T., Goshiro, Y., Kato, H., and Ikeda, S. (2017). EyeAR: Refocusable Augmented Reality Content through Eye Measurements. *Multimodal Technologies and Interaction*, 1(4):22:1–22:18.

Rosenthal, R. (1994). Parametric measures of effect size. In *H. Cooper & L. V. Hedges (Eds.), The handbook of research synthesis*, pages 231–244.

Shoemake, K. (1985). Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 245–254.

Shoemake, K. (1992). Arcball: A user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the Conference on Graphics Interface '92*, pages 151–156.

Simon, A. and Doulis, M. (2004). NOYO: 6DOF Elastic Rate Control for Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 178–181.

Slater, M., Perez-Marcos, D., Ehrsson, H., and Sanchez-Vives, M. (2008). Towards a digital body: The virtual arm illusion. *Frontiers in human neuroscience*, 2:6.

Smith, J., Vertegaal, R., and Sohn, C. (2005). Viewpointer: Lightweight calibration-free eye tracking for ubiquitous handsfree deixis. pages 53–61.

Song, J., Cho, S., Baek, S.-Y., Lee, K., and Bang, H. (2014). GaFinC: Gaze and Finger Control interface for 3D model manipulation in CAD application. *Computer-Aided Design*, 46:239–245.

Stellmach, S. and Dachselt, R. (2013). Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 285–294.

Stoakley, R., Conway, M. J., and Pausch, R. (1995). Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, volume 95, pages 265–272.

Takagi, M., Oyamada, H., Abe, H., Zee, D. S., Hasebe, H., Miki, A., Usui, T., Hasegawa, S., and Bando, T. (2001). Adaptive Changes in Dynamic Properties of Human Disparity-Induced Vergence. *Investigative Ophthalmology & Visual Science*, 42(7):1479–1486.

Tanriverdi, V. and Jacob, R. J. K. (2000). Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 265–272, New York, NY, USA. Association for Computing Machinery.

Taptagaporn, S. and Saito, S. (1990). How Display Polarity and Lighting Conditions Affect the Pupil Size of VDT Operators. *Ergonomics*, 33(2):201–208.

Theofilis, K., Orlosky, J., Nagai, Y., and Kiyokawa, K. (2016). Panoramic view reconstruction for stereoscopic teleoperation of a humanoid robot. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 242–248. IEEE.

Toyama, T., Orlosky, J., Sonntag, D., and Kiyokawa, K. (2014). A Natural Interface for Multi-Focal Plane Head Mounted Displays Using 3D Gaze. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 25–32.

Toyama, T., Sonntag, D., Orlosky, J., and Kiyokawa, K. (2015). Attention engagement and cognitive state analysis for augmented reality text display functions. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 322–332.

Uhlhorn, B. L. (2010). Display System Intensity Adjustment Based on Pupil Dilation. US Patent 7,744,216.

Van den Bergh, M. and Van Gool, L. (2011). Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72.

Vertegaal, R. (1999). The gaze groupware system: Mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 294–301, New York, NY, USA. Association for Computing Machinery.

Vidal, M., Bulling, A., and Gellersen, H. (2013a). Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 439–448.

Vidal, M., Bulling, A., and Gellersen, H. (2013b). Pursuits: Spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '13, pages 439–448.

Ware, C. and Mikaelian, H. H. (1986). An evaluation of an eye tracker as a device for computer input. *SIGCHI Bull.*, 17(SI):183–188.

Watson, A. B. and Yellott, J. I. (2012). A Unified Formula for Light-Adapted Pupil Size. *Journal of vision*, 12(10):12:1–12:16.

Winn, B., Whitaker, D., Elliott, D. B., and Phillips, N. J. (1994). Factors Affecting Light-Adapted Pupil Size in Normal Human Subjects. *Investigative Ophthalmology & Visual Science*, 35(3):1132–1137.

Wong, A. and Mirov, R. N. (2015). Apparatus and Method for Display Lighting Adjustment. US Patent 8,970,571.

Wu, H., Kitagawa, Y., Wada, T., Kato, T., and Chen, Q. (2007). Tracking iris contour with a 3d eye-model for gaze estimation. pages 688–697.

Yamazoe, T., Kishi, S., Shibata, T., and Kawai, T. (2009). LCD Backlight Control for Visibility of Monocular Head-Mounted Displays. In *Proceedings of the Industry Applications Society Annual Meeting*, pages 1–5.

Zhao, Y., Szpiro, S., and Azenkot, S. (2015). Foresee: A Customizable Head-Mounted Vision Enhancement System for People with Low Vision. In *Proceedings of the International ACM SIGACCESS Conference on Computers & Accessibility*, pages 239–249.