



TOHOKU UNIVERSITY
SENDAI, JAPAN

**DESIGN AND EVALUATION OF 3D INTERACTION
TECHNIQUES FOR SINGLE AND MULTI-USER DISPLAYS**

A dissertation submitted in conformity with the requirements for
the degree of Doctor of Philosophy

Graduate School of Information Sciences

Research Institute of Electrical Communication

Tohoku University

Author

Kasim OZACAR

Supervisor

Yoshifumi KITAMURA, Professor

AUGUST 22, 2016

Design and Evaluation of 3D Interaction Techniques for Single and Multi-User Displays

Author

Kasim OZACAR

Examination Committee:

- Yoshifumi KITAMURA, Professor
- Yôiti SUZUKI, Professor
- Satoshi SHIOIRI, Professor

Copyright Notices

Several sections of this dissertation have previously appeared in ACM, Oxford University Press, and IEEE publications. Permission from ACM, Oxford University Press, and IEEE have been granted for these works to appear in this dissertation. Each publication is listed below:

Chapter 3

Juan David Hincapié Ramos, **Kasim Ozacar**, Pourang Irani, and Yoshifumi Kitamura: GyroWand: An Approach to IMU-Based Raycasting for Augmented Reality, *Computer Graphics and Applications*, Volume 36, Issue 2, pp. 90-96, Mar.-Apr. 2016, IEEE. DOI=<http://dx.doi.org/10.1109/MCG.2016.21>

Juan David Hincapié Ramos, **Kasim Ozacar**, Pourang Irani, and Yoshifumi Kitamura: GyroWand: IMU-based Raycasting for Augmented Reality Head-Mounted Displays, In *Proceedings of ACM Symposium on Spatial User Interaction (SUI '15)*, pp. 89-98, August 2015. DOI=<http://dx.doi.org/10.1145/2788940.2788947>

Chapter 4

Kasim Ozacar, Juan David Hincapié Ramos, Kazuki Takashima, and Yoshifumi Kitamura: 3D Selection Techniques for Mobile Augmented Reality Head-Mounted Displays, 12 pages, To be appeared in *Interacting with Computers*, Aug. 3, 2016, Oxford Journals.

Chapter 5

Kasim Ozacar, Kazuki Takashima, and Yoshifumi Kitamura: Direct 3D Object Manipulation on a Collaborative Stereoscopic Display. In *Proceedings of ACM Symposium on Spatial User Interaction (SUI '13)*, pp. 69-72, July 2013. DOI=<http://dx.doi.org/10.1145/2491367.2491374>

Kasim Ozacar, Takuma Hagiwara, Jiawei Huang, Kazuki Takashima, and Yoshifumi Kitamura: Coupled-Clay: Physical-Virtual 3D Collaborative Interaction Environment. In *Proceedings of IEEE Virtual Reality Conference (VR'15)*, pp. 255-256, March 2015. DOI: 10.1109/VR.2015.7223392

ACM COPYRIGHT NOTICE: Copyright © 2013, 2015, 2016 by ACM. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permis-

sion and/or a fee. Request permissions from permissions@acm.org or Publications Dept., ACM, Inc., fax +1 (212) 869-0481.

OXFORD UNIVERSITY PRESS COPYRIGHT NOTICE: This is a draft of a chapter (Chapter 4) that has been accepted for publication by Oxford University Press in the forthcoming book *Interacting with Computers* by/edited by Ian Oakley due for publication in 2016.

IEEE COPYRIGHT NOTICE: ©2016 IEEE. Reprinted, with permission, from Kasim Ozacar, Pourang Irani, and Yoshifumi Kitamura, *GyroWand: An Approach to IMU-Based Raycasting for Augmented Reality*, *IEEE Computer Graphics and Applications*, Mar.-Apr. 2016. In reference to IEEE copyrighted material which is used with permission in this dissertation, the IEEE does not endorse any of Tohoku University's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

Abstract

There has been growing increasingly attention in interactive displays technologies from a wide range of fields such as entertainment, visualization, education, industrial design, and so on. Forms of displays can vary, ranging from 2D to 3D, each can also vary according to intended use such as for single and multi-user interaction. Furthermore, motion tracking systems with various approaches such as optical, inertial and magnetic trackers, provide users with mid-air input. Such systems ranges from tracking a rooms-size volume to tracking finger/hand, making such displays interactive in any forms, and enable users to interact through mid-air. While these technologies provide a wide range of design possibilities, designing relevant 3D mid-air interfaces is still challenging due to substantial dependencies regarding displays and tracking systems.

In this dissertation, I focus on 3D interaction issues with single and multi-user displays for three widespread settings: personal, collaborative, and public-exhibited displays. Recent head-mounted displays (HMDs), one of the most popular personal 3D display, are compact and particularly utilized in mobile settings; however, their major drawback is lack of powerful mid-air interaction techniques with spatial objects due to intrinsic problems of their low-cost built-in trackers. Therefore, I first investigated these trackers, addressed these problems, and designed promising 3D selection techniques for HMDs with their built-in trackers. Second, collaborative 3D displays such as holographic and perceptively-corrected 3D stereo displays allow multiple users to share and manipulate displayed content through various modalities for collaborative design, entertainment, and data exploration while maintaining face-to-face formation. I utilized IllusionHole as a feasible multi-user collaborative 3D display providing distortion-free, sharable, directly accessible stereoscopic views for multiple users around its tables-shape enclosure, and aimed to propose mid-air interaction techniques with shared 3D objects in collaborative tasks. The third exploration is on public-exhibited displays that have been increasingly deployed in various (semi-) public spaces for single and/or multi-user interaction, including galleries, schools and shopping centers. While motion-tracking technologies make these displays interactive and ready for public deployment, 3D mid-air interaction with such displays is still challenging due to novelty of the such interaction, which makes unclear to passers-by around the public display how to perform interaction.

The first exploration was on issues related HMDs. I proposed GyroWand in which accessible inertial measurement unit (IMU) sensing is utilized to bring raycasting selection techniques into mobile AR/VR, by addressing a number of challenges, such as identifying the source of ray and the ability to disambiguate the selection. Result of a series of studies suggests that using built-in IMU alone is a suitably good approach to adopt raycasting for 3D object selection in HMDs. After that, an empirical study was conducted to offer general

insights of 3D mid-air interaction techniques using built-in sensors in HMDs (i.e. gesture trackers, gaze tracking) where five promising selection techniques including GyroWand are validated. Results showed that IMU-based head controlled raycasting significantly assists 3D selection interaction for speed and user preferences.

The second work was regarding collaborative 3D displays, I newly designed a thin and compact, and flexible IllusionHole using a 3D TV. Also, the directness offered by IllusionHole led to create direct finger-based 3D interfaces, by which multiple viewers can directly point out, select and manipulate shared 3D content from their current point of view around the 3D display. Result of a user study showed that proposed finger-based direct interface provides faster content selection and manipulation than 3D cursor technique. The proposed direct interface also successfully enabled a group of users around IllusionHole to manipulate the virtual form of physical models to remotely contribute to 3D modelling tasks.

The third, regarding issues with public-exhibited displays, while touch interaction is common modality for these displays, mid-air interaction has several advantages such as being hygienic, allowing to interact with larger displays, and enabling interaction without come close to display. However, the most important challenge is that lack of design insight on how passers-by start interactions and how they use simple mid-air gesture sets in such displays. I addressed how to design mid-air interface for passers-by around such displays through an interactive game played with simple mid-air gestures in a public space, and collecting mid-air gestures to know how passers-by interact. I also analyzed effect of different types of conditions on interaction performances of passers-by. Results of the study showed that modelling, which demonstrates what movements the passer-by should make, was the most effective and efficient at improving interaction.

The primary contributions of this dissertation include a review of 3D displays and 3D mid-air interaction techniques, the implementation of a set of 3D selection and manipulation techniques, the design and evaluation of novel 3D mid-air interfaces for single and multi-user displays; and findings from several user studies. This dissertation offers deep insights on various 3D interaction techniques for these common types of displays, and a new basis for research into interaction techniques in 3D AR and VR.

Acknowledgment

First and foremost, I would like to thank my first advisor, Prof. Yoshifumi Kitamura, for providing his expertise and guidance, for his flexibility and understanding, and for keeping me on-track. He provided me with an excellent opportunity to study in his lab and enjoy life in Sendai.

Besides my advisor, I would like to thank the rest of my dissertation committee: Prof. Yôiti Suzuki and Prof. Satoshi Shioiri, for their encouragement, insightful comments, and some hard questions to improve my dissertation.

I also would like to offer my regards to the Assistant Prof. of my laboratory Kazuki Takashima, who helped me in completing this dissertation by exchanging interesting ideas, thoughts and opinions, and for the review of early drafts of my works.

Special thanks also to all my graduate laboratory friends; Jiawei Huang and Takuma Hagiwara for invaluable assistance. Dozens of people have helped and taught me. I wish to thank Dr. Juan David Hincapié-Ramos who have provided me with hands-on working experience, taught me many concepts in Human-Computer Interaction field, and programming. I have been very lucky to have so many people around.

I also want to thank the secretary of our laboratory, Nobuyo Maejima, for helping me especially to cope Japanese documents.

I would also like to convey thanks to the Republic of Turkey Ministry of National Education for providing the financial means and giving me the opportunity to study abroad in Japan.

Last but not the least, I wish to express my deepest gratitude to my wife and my sweetheart 2-year old daughter, my greatest friends, my parents and siblings for supporting me spiritually during my doctoral course and also for their love, patience, and encouragement.

Furthermore, I wish to thank all my friends, past and present, without all of them, this work would not have been possible.

Contents

<i>Abstract</i>	i
Acknowledgment	ix
1. <i>Introduction</i>	1
1.1 Interactive Single and Multi-User Displays	3
1.1.1 Head-mounted Displays (HMDs)	4
1.1.2 Collaborative 3D Displays	6
1.1.3 Public-Exhibited Displays	7
1.2 Objective	8
1.3 Outline	9
2. <i>Background Literature</i>	13
2.1 Introduction	13
2.2 3D Displays	13
2.2.1 Stereoscopic Displays	13
2.2.2 Auto Stereoscopic Displays	16
2.3 Interactive 3D Displays	17
2.4 Spatial Motion Tracking	17
2.5 Interaction in 3D	18
2.5.1 Human pointing models	19
2.5.2 Selection Techniques	20
2.6 Collaborative Displays	24
2.6.1 Co-located and Remote Collaboration	24
2.7 Summary of Background Literature	25
3. <i>IMU-based Raycasting for a Single-User Display</i>	27
3.1 Introduction	27
3.2 Related Work	29
3.3 Using IMU-data	31
3.3.1 Challenges Associates with IMU data	31
3.3.2 IMU-based Raycasting	33
3.4 On-body Raycasting Origin	36
3.4.1 Study 1: Raycast Origin	38
3.4.2 Results	41
3.4.3 Discussion	41
3.5 IMU-based Disambiguation	43

3.5.1	Lock and Drag	43
3.5.2	Lock and Twist	45
3.5.3	Auto-lock and Twist	46
3.5.4	Study 2: Selection Disambiguation	47
3.5.5	Results	49
3.5.6	Discussion	50
3.6	General Discussion	52
3.6.1	Implementation Details	52
3.7	Conclusion	53
4.	<i>Comparison of 3D Selection Techniques for a Singe-User Display . . .</i>	55
4.1	Introduction	55
4.2	Related Work	57
4.2.1	Sensing Technologies of HMDs	57
4.2.2	3D Selection Techniques	58
4.2.3	A Brief Survey of Selection Techniques	59
4.3	Selection Techniques for Mobile HMDs	60
4.3.1	Raycasting Techniques	61
4.3.2	Hand-controlled 3D Point Cursor Techniques	62
4.4	User Study: 3D Object Selection Task	64
4.4.1	Results	67
4.4.2	Discussion	71
4.4.3	Design Consideration	74
4.5	Conclusion	75
5.	<i>3D Selection and Manipulation Techniques for Multi-User Display . . .</i>	77
5.1	Introduction	77
5.2	Related Work	79
5.2.1	Interactive Multi-user Stereoscopic Displays	79
5.2.2	Mid-air Interaction	80
5.2.3	Collaborative User Interfaces for Co-located and Remote 3D Interaction	81
5.3	An interactive multi-user 3D display using a 3D TV	82
5.3.1	Principle	82
5.3.2	Trade-off relationship among design parameters	82
5.3.3	Movable volume for a user	85
5.3.4	Viewing Angle Between Users	86
5.3.5	Viewing Volume for a User	87
5.3.6	Maximum Number of Users	88
5.4	Implementation Details	88
5.5	Designing Direct Multi-finger Interface	90
5.5.1	Direct Multi-finger Object Manipulation	90
5.5.2	Sharing Object Manipulation	92
5.5.3	Discussion	92
5.6	User Study: Investigating Direct Mid-air Interface	93
5.6.1	3D Cursor	94

5.6.2	Experimental Design	95
5.6.3	Results and Discussion	96
5.7	3D Interaction for Co-located and Remote Collaborations	98
5.7.1	Virtual Interaction Space	99
5.7.2	Physical Interaction Space	100
5.7.3	Discussion	101
5.8	Conclusion	102
6.	<i>Effects on Guidance for 3D Mid-Air Interaction</i>	103
6.1	Introduction and Related Work	103
6.2	Methodology	104
6.2.1	Using game Mechanics for Data Collection	104
6.2.2	Defining an Atomic Gesture and Composite Gesture	105
6.2.3	Representing Gesture Signatures in a Relational Database	106
6.2.4	Creating a Gesture Recognition System	107
6.2.5	Designing the Game Concept and Scoring System	107
6.3	System Implementation and Deployment	108
6.3.1	System Architecture	108
6.3.2	Game Design	109
6.3.3	Deployment Venue Design	109
6.4	Experiment Design	112
6.4.1	Independent Variables	112
6.4.2	Dependent Variables	113
6.4.3	Experimental Protocol	115
6.4.4	Results	116
6.4.5	Discussion	118
6.5	Conclusions and Future Work	120
7.	<i>Conclusion</i>	123
7.1	Summary	123
7.2	Research Contributions	125
7.2.1	Contributions	125
7.2.2	Future Directions	126
7.3	Conclusion	128
8.	<i>Bibliography</i>	131

List of Figures

1.1	Single and multi-user displays	3
2.1	Classification of Display Technologies	14
2.2	Anaglyph 3D. Adapted from [42]	14
2.3	Polarized-filtered 3D. Adapted from [42]	15
2.4	Time-multiplexed 3D. Adapted from [42]	15
2.5	Head-mounted 3D. Adapted from [42]	16
2.6	Universal tasks in 3D interaction	18
2.7	Fitts' Law	20
2.8	Classification of selection techniques by task decomposition . . .	21
2.9	Classification of interaction techniques by metaphor	21
2.10	Co-located and remote collaborations	24
3.1	GyroWand enables raycasting in HMDs	28
3.2	Pronation and supination	34
3.3	The state machine for IMU-based raycasting	35
3.4	On-body locations for the GyroWand's ray origin	37
3.5	Targets	40
3.6	Quantitative results	42
3.7	Lock&Drag	44
3.8	Lock&Twist	46
3.9	Autolock &Twist	47
3.10	Three obstacle walls and one target.	48
3.11	Quantitative results	51
4.1	Ray-based techniques: Head cursor and GyroWand	61
4.2	Hand-gesture based techniques	63
4.3	Recognition state machine for flick gesture.	64
4.4	A participant interacting by using an HMD	65
4.5	Rendering of user's view	66
4.6	Quantitative results	68
4.7	First-last arrival time to target	70
4.8	Borg RPE score	70
4.9	Amount of angular displacements	72
5.1	Remote collaboration	81
5.2	Generation of stereo-images and image drawing areas	83

5.3	Trade-off among parameters	84
5.4	Design parameters of the IllusionHole	84
5.5	Movable Volume for a User	85
5.6	Viewing Angle between Users	87
5.7	Viewing Volume for a User	87
5.8	Users' viewpoint	89
5.9	Configuration of new IllusionHole	90
5.10	Mid-air selection and manipulation tasks	91
5.11	Sharing direct object manipulation	92
5.12	3D Cursor	94
5.13	Interaction time and scaling error	96
5.14	Position error and scaling errors	96
5.15	General Feelings	96
5.16	Overview	99
6.1	Composite Gesture	106
6.2	System Architecture	108
6.3	Illustrated sequence of game dynamics	110
6.4	Scoring	110
6.5	Public Exhibition	111
6.6	Exhibit venue	111
6.7	Passers-by playing the game.	112
6.8	Experiment conditions from Week 1 to Week 4	114
6.9	Adherence vs. Condition	117
6.10	Graph of Correctness vs. Condition.	117
6.11	Graph of Score vs. Condition	118

1

Introduction

The history of 3D display technology draw the way back to start of photography. In 1849, David Brewster invented "lenticular stereoscope", which is the first portable 3D viewing device. After years, a stereo animation camera was invented, followed the first anaglyph 3D, which encodes each eye's image using filters of different colors one for each eye. When viewed through glasses with color-coded lenses, each of the two images reaches the eye it's intended for, introducing a stereoscopic image. In 1968, Sutherland presented the first head-mounted three dimensional display [166]. The main idea behind the three-dimensional display is to provide perspective 2D images which change as the user moves. A mechanical and an ultrasonic sensor are used to track the position of the user's head. In contrast to today's 3D stereo visualization, this first system provided users to 2D imagery. Stereo viewing with motion parallax provides improved user performance in 3D visualization. Ware et al. [189] proposed a fish tank Virtual Reality(VR) system in providing the user with a more realistic visualized 3D data.

Numerous types of 3D displays have emerged since the development of the first 3D display: Stereoscopic displays, Auto-stereoscopic displays (multi-view displays, volumetric displays and holographic displays). Volumetric displays do not provide directly to merge their hands or finger to directly interact with 3D content. The volumetric displays generate true volumetric 3D images by actually rendering points in physical 3D space, typically have a 360-degree field of view and do not require for users to wear and head-track device and glasses. Current volumetric displays [47] have a physical enclosure around the display, which does not allow users directly reach in and grab even within-arm's reach objects. Although holographic 3D displays are considered to be true 3D

display technology, its implementation requires many technical challenges. For instance, a holographic displays require a pixel size at nanometer scale, resulting in a zillion pixels on a reasonable display screen [42]. Multi-viewer displays enable tracking of the viewing field by tracking users eye position so that pair of images is displayed to the users with both horizontal and vertical parallax [176]. In auto stereoscopic multi-view displays, large Fresnel and lenticular lenses, parallax barriers, and prisms are used to control light path and to form exit pupils, resulting in loss of brightness and loss of spatial resolution [176]. Note that from now we will use the term of "3D displays" for "3D stereoscopic displays".

Visualization, entertainment, media industry, collaborative industrial designing, and signage & advertising for public environments have been evolved due to rise in interactive 3D content. Increasing adoption in 3D content grows up 3D display in global market. Among all types of 3D displays including volumetric, 3D stereoscopic and HMD; the stereoscopic displays and HMDs are most preferred types of 3D displays due to their applications in PCs and smartphones. Moreover, 3D positional trackers enable users to build 3D interaction between physical and virtual environments, providing direct or indirect 3D mid-air interaction with displayed content. Mid-air interaction with finger-pointing or hand-held device-pointing techniques is an interesting alternative due to the more direct interaction way than typical 2D touch interaction. The main reason is that while touch interaction is standard for touch displays, this interaction requires mapping 2D input space to 3D virtual space when using it for 3D user interfaces, resulting difficulties in manipulation of 3D objects [103]. Currently, there are various types of motion tracking approaches such as optical, magnetic, inertial, acoustic, and mechanical. Among all, inertial and optical sensing have increasingly interest in due to cost and wide usage scenarios. Thanks to such tracking capabilities, virtual reality (VR) experiences increasingly become more powerful, popular, and deployable for various interactive scenarios and users. Consumer demands for interactive 3D displays is steadily increasing and expected to rise. This rise motivated the need to bring 3D interaction opportunities for such devices.

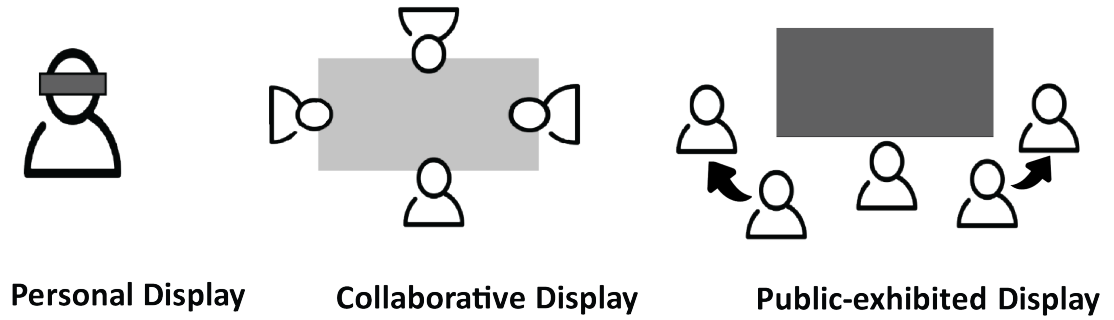


Fig. 1.1: An illustration of single and multi-user displays for personal and collaborative interaction, and for interaction in public-exhibited space

1.1 Interactive Single and Multi-User Displays

3D Stereoscopic displays provide the users to perceive depth effect leverage them to visualize the data by increasing dimensionality in a three-dimensional space, and have been mainly used in virtual reality (VR) and augmented reality (AR) for a wide spectrum of everyday life application areas such as entertainment, training, 3D design and, medications. Unlike to typical 3D displays, which are even take place in our home, interactive 3D displays provide interaction between the users and virtual content screened on the display. Tracking users' eye position, provides depth information to be created and screened on display.

Moreover, tracking user's fingers and hands allow the users to directly interact with the virtual objects to select and manipulate them. Researchers have proposed and evaluated variety of 3D interaction techniques. One category of such techniques offer indirect interaction for remote objects, such as ray casting [132] [133]; and other techniques[111][130] utilize human hand directly in the virtual environment offers direct manipulation. This type of direct interaction is called as *direct manipulation interface* [67]. Using a direct manipulation interface provides perceiving spatial relation among the object in the display. There are many types of technologies to provide stereoscopy, varying from single anaglyph glasses to shutter glasses. They have a large variety of form and support a wide range of interaction techniques and can be used by a single and/or a variable number of users. While personal interactive displays allow the user to interact with the virtual 3D content, collaborative displays provide multiple users to work together on a task and also interact each other during the collaboration. Among all types of 3D displays which includes volumetric, stereoscopic and head mounted displays (HMDs); the stereoscopic displays and HMDs should remain the most preferred type of 3D display due to

their applications in TVs, smartphones, and PCs. Nowadays, one of the most common personal type 3D displays is head mounted displays (HMDs) because HMDs market is expect to grow with fastest compound annual growth rate of 49.9% during 2014-2020 [26].

1.1.1 Head-mounted Displays (HMDs)

Head mounted displays (HMDs) are one of the most popular 3D stereoscopic displays. HMDs are attached to the user head, enables 3D content for personal use, reflects projected image pairs from an optical system to each eye, and has a sensing system to provide users a dynamic point of view. HMDs can be classified as immerse and see-through.

While Immersive types fully focus on virtual environment; thus isolating the users from real-life, see-through HMDs allow users to see-through it. Recent HMDs are compact and particularly used in mobile settings. Because, new generation 3D HMDs contain low-cost tracking technologies that allow such devices to interpret users' motion in the real world in relation to the virtual content for selection and navigation tasks. These low-cost technologies have potential to enable AR and VR applications to take place outside the research laboratories at an affordable price without the need of fixed tracking infrastructure, . However, the major challenge is lacking of powerful mid-air interaction techniques for intuitive mobile interactions with objects in the 3D space by its limited built-in or compact tracking sensors (optical and motion sensing).

Therefore, my first exploration is to use accessible inertial measurement units(IMU) sensing in order to bring a common interface into mobile HMDs. On the other hand, raycasting, an interaction technique widely explored in traditional virtual reality (VR) and experimental results show that raycasting outperforms other 3D interaction techniques in selection tasks 2[21].

Research Question: How to enable 3D selection techniques by IMU-based raycasting for mobile interaction in HMDs

Using raycasting, a ray projecting away from the user's arm and into a scene, as an interface tool has been widely used in virtual reality domain. However, many of raycasting interfaces require absolute trackers, thus, a new raycasting approach is needed to enable interaction in mobile context. For this reason, I aimed to address a number of challenges, such as identifying the source the ray, noise handling, and the ability to disambiguate the selection with my novel raycasting approach.

In Chapter 3, I present GyroWand, a raycasting technique for 3D interactions in self-contained Augmented Reality (AR) head-mounted displays. Unlike traditional raycasting which requires absolute spatial and rotational tracking of a user's limb, GyroWand relies on the relative rotation values captured by an inertial measurement unit (IMU) on a handheld controller. However, these values cannot be directly mapped to the ray direction due to the phenomenon of sensor drift and the mismatch between the orientations of the physical controller and the virtual content. To address these challenges my approach 1) interprets the relative rotational values using a state machine; 2) handles drift by resetting initial rotation; 3) does not initiate raycasting from the user's hand, but rather from on-body spatial coordinates; and 4) provides IMU-based disambiguation mechanisms. I evaluated the performance and convenience of different GyroWand design parameters, and presented a summary of the lessons learned for the adoption of raycasting in mobile augmented reality head-mounted displays.

Also, investigating how different 3D interaction techniques, which using different built-in tracking technologies apply to the unique circumstances of HMDs, in mobile scenario is very crucial to reveal the best technique due to display configuration and characteristic in such novel devices. Moreover, type of built-in tracking sensing can have a major impact on 3D interaction performance. Therefore, it is important to understand the capabilities of current built-in display and sensing technologies on these devices, and to determine how they influence selection performance and user preference. I explore 3D interacting techniques, which rely solely on tracking built-in tracking technologies in commercially-available HMDs for 3D selection tasks.

Research Question: What techniques will make the selection task of 3D objects easier in HMDs for mobile interaction

Selection techniques, which solely rely on built-in tracking technologies (i.e. gesture trackers, gaze tracking, and inertial measurement units), should be developed to reveal the best technique in HMDs for mobile interaction.

In chapter 3, a raycasting technique was successfully adopted such that brings 3D interaction to mobile scenarios. In order to answer above mentioned question, I explored more selection techniques, and conducted a user study in Chapter 4 evaluating five selection techniques for HMDs. All techniques have

been explored are designated to support mobile usage scenarios in which the devices do not need external tracking tools or special environments. Therefore, the selected techniques rely solely on tracking technologies built into conventional commercially- available 3D HMDs (i.e. gesture trackers, gaze tracking, and inertial measurement units). While two techniques are based on raycasting using built-in inertial measurement unit (IMU) sensing, three techniques are based on a hand-controlled 3D cursor using gestural tracking. These techniques were compared through an experiment with twelve participants. The results showed that controlling raycasting with head orientation (i.e. IMU on the headset) was the fastest, fatigue-less, and the most preferable technique to select spatially arranged objects. I discussed the implications of findings for designing interaction techniques in mobile HMDs.

1.1.2 Collaborative 3D Displays

3D stereoscopic displays have many advantages such as enabling multiple users to simultaneously view the same 3D content from their perspectives, and therefore, delivering collaboration through such displays. In addition to providing simultaneous observation of stereoscopic image pairs, an ideal multi-user stereoscopic display should have interface elements which must be accessible and able to handle parallel input from all the users for collaborative tasks. Furthermore, an interactive tabletop display should provide users face-to-face interaction with verbal or visual communication and allow them to share the awareness regarding collaborative task. With spreading high-speed network during the recent years, interactive tabletop displays do not only enable users to interact with them in collocated study, but also allow to interact with remote users anywhere around the world. This remote interactive collaboration allows both local and remote partners to modify the properties of the shared content, a product prototype, 3D interactive museum, and 3D art. Consider a such scenario in which a group of designers works on a physical object, and wants to allow remote users to observe it and to join in any phase of the design. Even more, enabling remote users to modify the 3D representation of designed model, and transfer their modifications on virtual models to the physical objects. With such cooperative capabilities, remote collaborators can directly influence the details of new designs during the design process as if they were co-located. In order to realize such collaborative scenario, an interactive collaborative environment must be designed.

In multi-user stereoscopic displays, each user wears a tracking sensor or

marker, which provide the system to use true position and orientation of stereo image pairs for the users. This provides a correct image pairs to each user according to their current positions. Otherwise, motion sickness results distortions when users move their head from side to side due to the fact that image perspective remains the same. Interface elements are another instrument in multi-user stereoscopic display, which must provide all users to directly interact displayed objects freely from any point of view. IllusionHole [78] is a powerful interactive multi-user supported stereoscopic display which fit the mentioned requirements. It provides a distortion-free 3D content at exactly the same position from each of the viewpoint of multiple users; therefore, it can be used in many cooperative work. However, the display configurations of the system have bulky components under the table, such as either a huge 3D projector or a set of projectors for stereo-view with a mirror for bottom-projection. These cause the system to have a big enclosure where users should interact in standing postures, hindering a flexible design. Also, traditional mouse and keyboard can be utilized to interact with such displays; however, it is problematic because the viewers want to move about in front of the displays because such large displays address multiple viewers. Therefore, I focused on research question in below:

Research Question: How to enable a group of users to directly select and manipulate 3D objects for collaborative tasks (e.g., modelling and visualization) from their own point of view around the IllusionHole?

Interacting directly with 3D content using 3D mid-air gestures seems a natural choice for IllusionHole. Therefore, I focus on this research question in Chapter 5.

1.1.3 Public-Exhibited Displays

Displays in public spaces play an important role in way we discover, and act in our environment by displaying visual form of information such as venue information, points of interest, billboards, art, and so on. Such displays have been increasingly deployed in various public spaces, including galleries, schools, and shopping centers. Although touch interaction is mainly used in these devices as common modality, 3D mid-air interaction has several advantages for interaction. For instance, it is appropriate for hygiene, enable passers-by to interact with very large displays deployed higher position, and enables interaction

without come close to display [184]. While motion-tracking make these displays interactive and ready for public deployment, there is a lack of design insight on how passers-by start mid-air interactions and how they use simple mid-air gestures in such displays. The reason is that using 3D mid-air gestures when interacting with such displays in public spaces is problematic due to lack of physical affordances, which provide clues to interaction, and leads to difficulties in learning regarding how to interact. Mid-air interaction with public-situated display introduces several challenges [184]. For instance, due to novelty of mid-air gestures in such displays it is unclear for passers-by how to initiate the interaction. Also, users don't know which gestures are available and how to execute such gestures. Another challenge is passers-by may give up if they do not understand immediately, therefore, do not succeed with their interaction [101].

Research Question: What kinds of guidance in public-exhibited displays make ease learning of mid-air interaction for passers-by?

I aimed to provide a set of guidances to passers-by for learning mid-air interaction immediatly around such displays by designing an interactive game, which is played with simple mid-air gestures in a public space, and collecting mid-air gesture data to know how passers-by interact with public-exhibited displays. I also analyzed the effect of different types of guidance on mid-air interaction performance of passers-by. , I focus on this research question in Chapter 6.

1.2 Objective

The main objective of this dissertation is to design and evaluate 3D interaction techniques for single and multi-user displays such as personal, collaborative, and public-exhibited displays. A set of 3D techniques for interacting with this displays are designed and evaluated through a series of user studies. This dissertation can be divided into three main parts, and each part aims to address a set of requirements regarding interacting with single-user and multi-user displays.

Firstly, possibility of using accessible sensing (i.e., IMU in smartphones), which are not readily available for applications in mobile contexts, on single-user head-mounted displays (HMDs) needs to be explored to bring into mobile AR a common interface for interacting with objects in 3D head-mounted displays.

Also, potential of promising both 3D virtual pointing and virtual hand techniques in virtual reality field must be explored to reveal and adopt the best technique among them for HMDs with accessible built-in sensing. Chapter 3 and Chapter 4 aims to address these requirements for 3D object selection tasks in HMDs.

Secondly, a multi-user display for collaborative tasks needs to have a simple configuration, compact and flexible design, and should enable users to directly interact collaboratively with 3D objects in mid-air, as well as, enable multiple users to share the real-time object manipulations from their individual point of view for various tasks such as modeling or visualization. Chapter 5 aims to address these requirements for a multi-user display.

Finally, public-exhibited single and/or multi-user displays lack of physical affordance, which makes mid-air gestural interaction highly challenging and hard-to-learn for passers-by. Therefore, effect of various types of guidance, help users to perform 3D mid-air gestures, need to be explored in terms of supporting passers-by learning how to perform the gestures. Chapter 6 aims to address these requirements for single and multi-user interaction with public-situated display.

1.3 Outline

In this dissertation, firstly I aimed to identify the properties and variates of single and multi-user display technologies. I will be built up the groundwork for various type of future interactive displays and interaction scenarios to build on.

In Chapter 2, I outline various previous related work which this work will build on.

In Chapter 3, I explored interaction issues for one of popular single user displays, optical head mounted displays (HMDs). I explore how 3D selection techniques can be integrated into high-level user interfaces for HMDs using built-in low-cost tracking technologies. Because these novel self-contained HMDs cover a wide range of applications, I aimed to bring raycasting into these devices for mobile interaction scenarios in Chapter 3. I reported the results of two user studies, in which I address a number of challenges, such as handling sensor noise and drift, identifying the source point of the ray and the ability to disambiguate the selection using ray casting interface. This studies contribute adoption of raycasting-based selection for stereoscopic HMDs.

Motivated by the results obtained in Chapter 3, I explored more interaction techniques that can be realized using not only built-in movement sensing but

also optical sensing for 3D selection tasks in mobile interaction scenarios in Chapter 4. I compared the fundamental performances of typical and promising virtual pointing and virtual hand techniques for 3D selection tasks in head mounted displays with built-in sensing (i.e. hand-held ray, natural hand gesture). Results showed that head controlled raycasting using IMU sensing (head cursor) facilitates the 3D selection task for speed and user preferences. This study shows that head cursor is faster, more accurate and more preferable way than the technique proposed in Chapter 3, and contributes adoption of IMU-based head cursor for 3D selection task in self-contained 3D head-mounted displays.

In Chapter 5, I focused on interaction issues regarding multi-user 3D Displays. IllusionHole [78] is one of such displays that provides a distortion-free 3D content at exactly the same position from individual viewpoints of multiple users; however, the display configurations of the system have bulky components under the table, resulting inflexible interaction environment. In Chapter 5, I designed and implemented a novel thin and compact interactive multi-user 3D display using a 3D TV for a typical collaborative workspace where users interact in a more flexible way. I also implemented a finger-based direct manipulation interface to the multi-user display system, in which multiple users share the real-time object manipulations from their own point of view. This study also contributes an empirical comparison of proposed direct interaction interface to an indirect 3D cursor method. I then utilized this direct interface to create a bi-directional remote collaborative interaction environment for co-located/remote modeling tasks. In the modeling task, a physical interactive workspace enables a user to directly build and modify a fully tracked physical object, a virtual interactive workspace enables a group of users to view the object and interact with it in real-time through its virtual model in the 3D tabletop display.

A 3D mid-air gesture-based system may be difficult for users to learn due to the lack of physical affordances providing clues to its operation. In Chapter 6, I focused on issues regarding learning how to mid-air interaction with public-exhibited displays . In order to make the system more learnable, certain guidance could be used. However, the question of which guidance to use, and why, is still an open question. To address this problem, I designed an interactive game, which is played with simple mid-air gestures, and exhibited in a public space. The interactive game served to collect mid-air gestural data to report how passers-by perform mid-air gestures on public-exhibited display. I also analyzed effect of different types of guidance on learning performance of passers-by. The results of the study showed that a short animated video, which

demonstrates what movements passers-by should do, was the most effective and efficient at improving mid-air interaction.

In chapter 7, I summarized the scope of works presented in this dissertation, listed main contributions, and discussed possible future directions.



2

Background Literature

2.1 Introduction

In this chapter, I review the previous research which will help to better understand 3D displays and interactions to build novel 3D interactive displays and interaction with 3D displays.

2.2 3D Displays

In this section I provide a background on existing 3D display technologies which present 3D data. These displays utilize depth cues, as explained in Chapter 1. I first discuss the display technologies and utilized the depth cues which the displays provide.

2.2.1 Stereoscopic Displays

Stereoscopic displays are effective tools for viewing and interacting with 3D content, and are still the most commonly utilized display type for 3D tasks. These binocular stereoscopic displays techniques requires user to wear a special glasses, by which he see slightly different two image pairs in each eye. There are many ways to provide such stereo image pair deliver. I would like to briefly introduce each techniques.

Anaglyph 3D

Anaglyph 3D techniques require a user to wear color-coded glass. Images for each eye is filtered with chromatically opposite colors(Such as red and cyan,

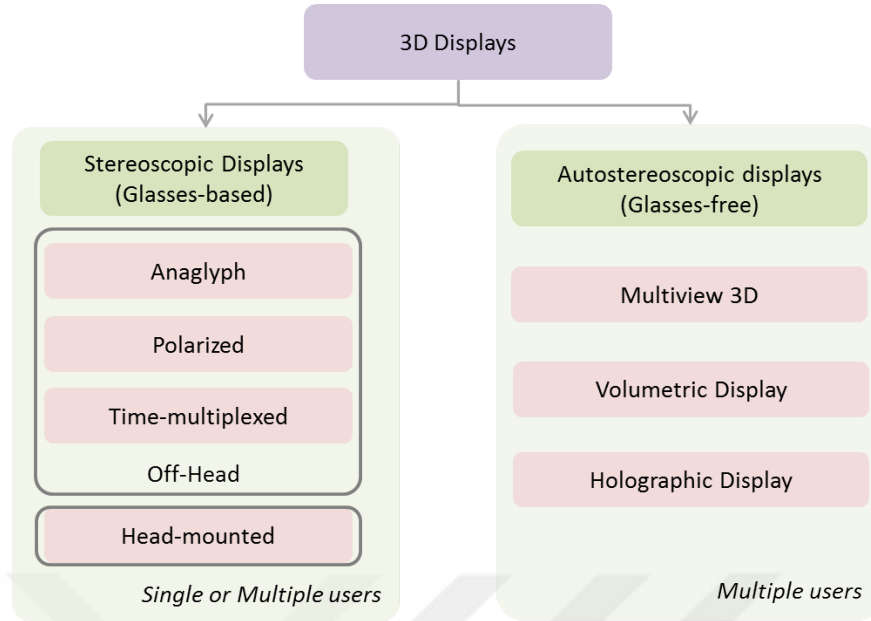


Fig. 2.1: Classification of Display Technologies

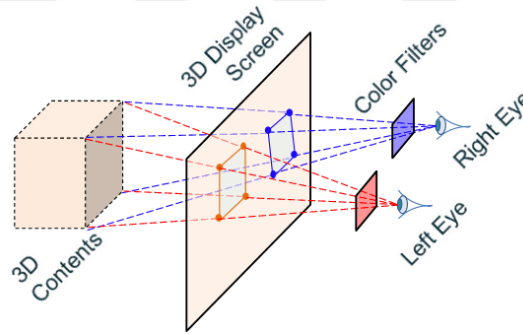


Fig. 2.2: Anaglyph 3D. Adapted from [42]

red and green.) 2.2. ColorCode 3-D is an anaglyph 3D stereo viewing that deployed in the 2000s by Sorenson et al. [156]. As can be seen in Figure 2.2, left eye receives the an image of nearly full color information and, right eye, a monochrome image to give the depth sensation.

Polarizer-filtered 3D

Polarizer-filtered glasses in combination with orthogonally polarized images presented on two displays, resulting each eye sees one image pair from one object (See Figure 2.3). Stereo projection screens must preserve polarization. Even though filters dissolve over sixty-percent of the light emitted, polarization techniques are widely used in viewing stereoscopic projection. LC-based, direct-view displays, and overhead panels are commercially available polarizer-filtered glasses types. [34].

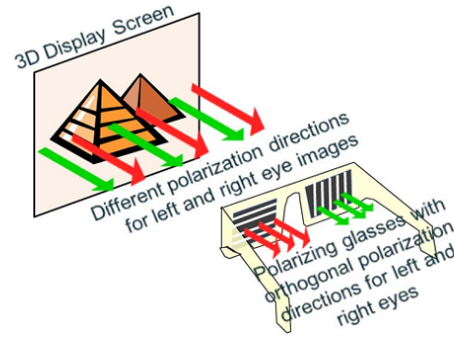


Fig. 2.3: Polarized-filtered 3D. Adapted from [42]

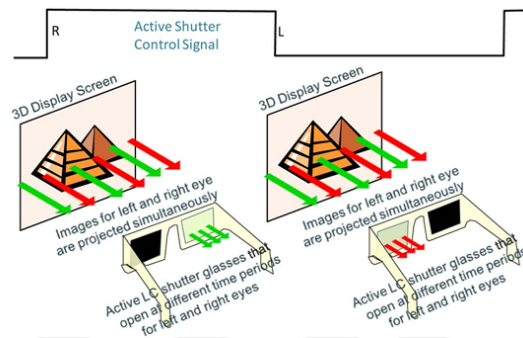


Fig. 2.4: Time-multiplexed 3D. Adapted from [42]

Time-multiplexed 3D

The displays using these techniques rely on capability of human visual system to merge the constituents of a stereo pairs across a time lag of up to 50ms [158]. The right and left-eye views are shown in rapid alternation and synchronized with a LC shutter, opening in turns for one eye, while occluding the other eye (See Figure 2.4). This shutter system is controlled via an infrared connection. When the viewer moves away from the display, shutters are switched to be transparent. Therefore, these types of displays compatible with 2D representation. Both image pairs are reproduced at full resolution by a single display, avoiding geometrical and color differences [158].

Head-mounted Displays

Head-mounted displays (HMDs) require a viewer to wear a display on the head, a helmet or a glasses that displays a separate video source in front of each eye to provide stereoscopic view (See Figure 2.5). These displays typically have two small LCD or OLED displays with magnifying lenses for each eye [23]. Progressive design can improve the performance while reducing the size of HMDs [28]. Nowadays, HMD technologies are used to show 3D stereo

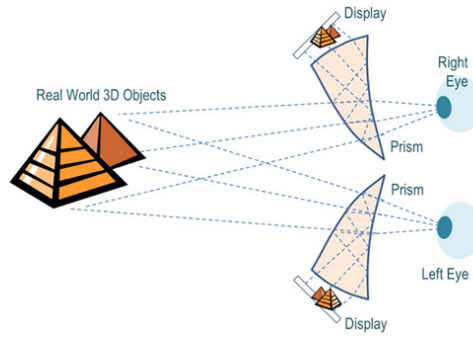


Fig. 2.5: Head-mounted 3D. Adapted from [42]

movies and images, to create interactive games, and virtual environments. HMDs with motion tracking sensing enable the user to look around the virtual world through his head movements, without needing a separate controller. By utilizing tracking system with 6-Degree-of-Freedom (move and rotate), the viewer can move within the scope of tracking, the view may move about within the limitations of the tracking system used. Currently, with emerging rapid advancements in computer graphics and the continuing reduction of hardware size, these devices are becoming available for end users at reasonable price. Head-mounted displays (see-through HMDs) may offer user experience by superimposing a virtual information layer on top of the real-world, creating augmented reality.

2.2.2 Auto Stereoscopic Displays

Auto stereoscopic displays can be divided into three types (2.1): Multiview 3D Display, volumetric display, and holographic displays. Multi-view 3D displays address the accommodation–convergence issue of stereoscopic displays. A multiview autostereoscopic 3D display is able to produce different views for different angular positions without requiring any special eyeglasses, providing both stereo and motion parallax depth cues to its viewers. In contrast to multi-view 3D displays that provide different 3D view in different angular positions to its viewers, volumetric 3D displays can display volumetric 3D images in real 3D space. In addition, holographic 3D displays allow the light wavefronts scattered from an object to be recorded and later reconstructed so that an eye can see a 3D image of the object even it is no longer present [42]. In general, auto stereoscopic displays are still cutting edge technology and are not readily available for common use. Therefore, these displays remain out of scope of this dissertation.

2.3 Interactive 3D Displays

Tracking the position of user's fingers/hand, and head or using a stylus enables the user to interact with virtual object presented on a stereoscopic display, which called as direct manipulation interface [67]. Interaction with stereoscopic data is a challenging task. I have discussed the existing main 3D display technologies, and evaluations have been done on them. In the remaining sections in this chapter, I will discuss issues related to interaction with this displays.

2.4 Spatial Motion Tracking

Interactive 3D stereoscopic displays do not only visualize 3D data, also provide a new dimension for interaction. Therefore, there have been numerous research focusing on spatial tracking sensing which provide three or more degrees of freedom (DOFs) to interact with 3D displays.

The control of a virtual camera in 3D environments requires 6DOFs, which can be directly controlled by 6-DOF input devices, like the Scene-in-Hand, Eyeball-in-Hand and Flying Vehicle metaphors proposed by Ware and Osborne [192]. In the Scene-in-Hand technique, movements of the user's hand are directly linked to corresponding movements of the scene. In the Eyeball-in-Hand technique, the user's hand serves as a virtual camera. In a qualitative oriented study, Ware and Osborne concluded that the different metaphors are best suited depending on a particular task.

Recently, image-based tracking systems have become very popular. As with optical systems, they use infrared light and a depth camera to get a single depth image of the view. Such systems resolve the spatial positional data of the tracking object by applying an image process algorithm (e.g., Kinect [151], Leap Motion) to the depth image. Other methods based on RGB images have also been proposed. [150] proposes an approach to wear video cameras on human body and compute the motion based on the captured video, however, it requires large amount of computation. Shiratori et al. [185] proposed a real-time method for tracking hand motion with a high accuracy that requires users to wear a colorful glove to track the whole hand. However, tracking a colorful glove requires a simple background to avoid conflict with the background colors. Even though Wang et al. [186] achieved a high-level motion capture result with the dynamics from bare-hand video data, this method cannot provide real-time tracking yet. Based on their algorithm, accurate identification (i.e., recognizing

each specific user by ID) requires much additional effort. For dexterous motion tracking and 3D interaction sensing, their lack of identification is critical.

Optical systems, which are the most widely used for many areas, such as VR and full-body motion capture, usually consist of multiple infrared cameras with an emitter and markers made of a reflective material. The cameras are arranged to completely observe the tracking space from different aspects. After the reflected light from the markers is captured by the cameras, the system triangulates their positions from the camera images. In addition to passive markers, some systems use active markers which are lit by synchronization with the cameras. However, with a single marker, this system provides only 3D positions and cannot identify multiple markers. The identification of multiple tracking points often requires such somewhat awkward approaches as a unique combinational use of several markers (e.g., 3-point triangle layout) for each tracking point. Since such schemes inherently use light, sometimes the targets cannot be continuously tracked because they are easily occluded from the cameras.

2.5 Interaction in 3D

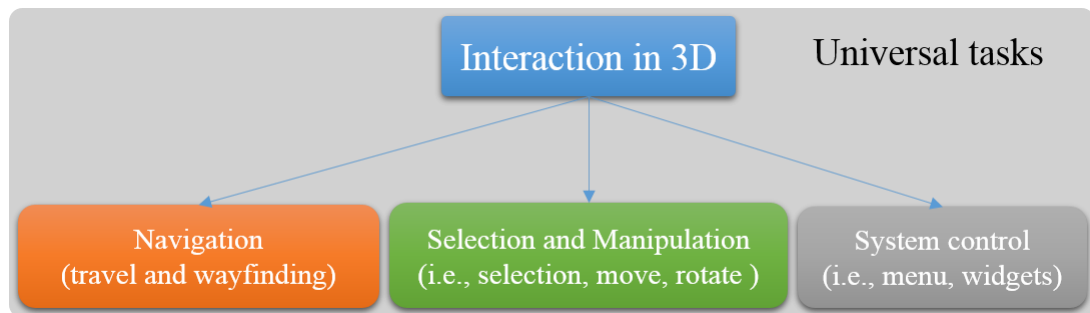


Fig. 2.6: Universal tasks in 3D interaction

There is a wide range of techniques to interact with 3D contents in virtual environment. These interactions can be classified according to three universal tasks: Navigation, selection and manipulation, and system control.

Navigation - Refers to the action of moving a user's point of view through a virtual environment. Navigation consists of two tasks: Travel and wayfinding [21].

Selection and Manipulation - The following tasks can be designated as basic manipulation tasks: positioning, scaling, and rotation.

- *selection* - refers to acquire a particular object from a set of objects in a virtual environment. A selection task is also called as a target acquisition task [205]. Real world equivalent of a selection task is picking an object with a hand/tool.
- *positioning* - refers to change the 3D position of an object in a virtual environment. Real world equivalent of a the selection task is moving an object from a location to another one.
- *rotation* -refers to change the orientation of an object. Real world equivalent of a rotation task is rotating an object to change its appearance.

System Control - In this dissertation, I only review previous work in *3D selection and manipulation techniques* for interactive virtual environments. I will review major 3D interaction techniques for 3D selection tasks, without considering indirect selection techniques (e.g. selecting from a menu or semantic selection).

2.5.1 Human pointing models

In order to improve user pointing performance for task completion time and number of errors, a number of approaches have been proposed. A common approach is to use human pointing models such as Fitts' Law [38] and the optimized initial impulse model. While Fitts' Law estimates the time required to acquire a target, the optimized initial impulse model suggests corrective sub movements toward a target.

Fitts' Law

Fitts' Law is a one-dimensional descriptive model of human pointing movement targeted two-dimensional selection tasks. It explains speed-accuracy trade-off characteristics of human movements [96]. Today's graphical user interfaces are designed respect to Fitts' Law.

Fitts' Law can be imagined with the Figure 2.7. $T = a + b \log_2 \left(1 + \frac{D}{W} \right)$ where T is time, a and b are empirical constants, D is distance and W is width of the target. Fitts' Law also defines the Index of Difficulty (ID) term, $ID = \log_2 \left(1 + \frac{D}{W} \right)$, in order to measure the difficulty of a task. Fitt's Law basically expresses that a graphical user interface must minimize the distance toward objects to be selected (target), and the target must be as wide as possible by taking display

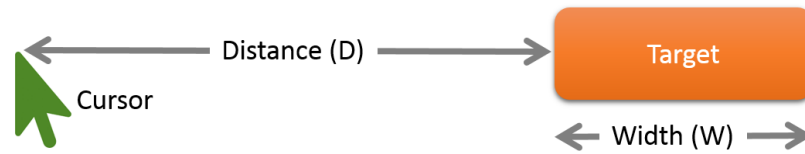


Fig. 2.7: Fitts' Law

constraints into account. There have been numerous studies that extended Fitts' Law to 2D [96][202] and 3D pointing tasks [137][64][24] with 2D and 3D target arrangement respectively. However, Fitts' Law only considers the pointing speed according to target size and its distance from the cursor. Fitts' Law does not provide any information regarding the performance of a limb or a device during interaction.

Two-component Model

In order to understand and analyze movements of limb or hand-held device during a pointing task, there is a need to understand the two-component model by Woodworth [198]. This model divides a goal-directed movement into two phases: Ballistic phase and correction phase.

The ballistic phase is the initial part of the movement, relatively fast and inaccurate, towards a target. In other words, the ballistic phase is designated to arrive at a target as quick as possible regardless of accuracy. If the target is not acquired, due to overshoots or undershoots, iterative slow and more accurate movements are performed until the target is acquired in the next phase, which is called correction phase.

2.5.2 Selection Techniques

Figure 2.9 presents the 3D selection and manipulation techniques by metaphors. Firstly, the techniques categorized by exocentric and egocentric metaphors. Exocentric techniques provide "a second dynamic viewport onto virtual environment" [139]. World-in Miniature is example for exocentric metaphor in which one can interact with virtual environment through a 3D viewport [162].

In Egocentric techniques, a user interacts as if he/she was in virtual environment. Egocentric techniques are divided into virtual hand and virtual pointer metaphors. , While users reach and grab the objects through a virtual hand in virtual hand metaphors, virtual pointing metaphor enable users to interact with objects by pointing at them.

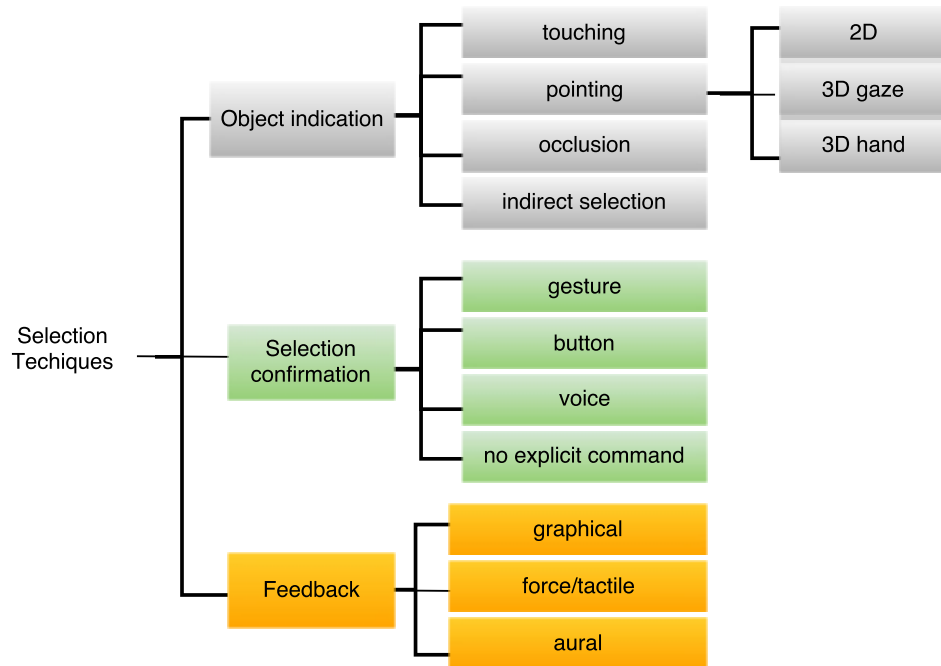


Fig. 2.8: Classification of selection techniques by task decomposition [20]

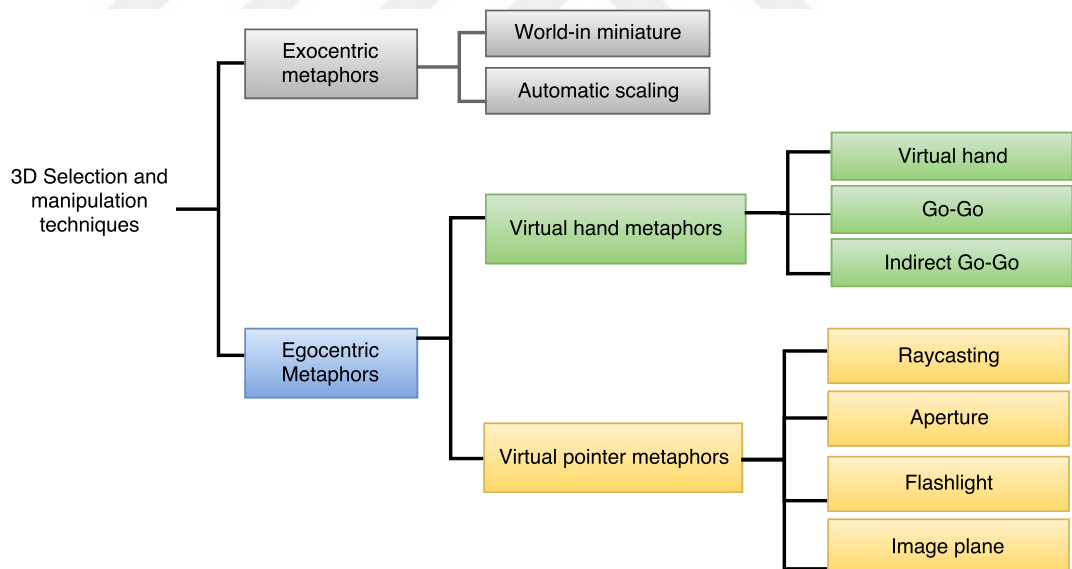


Fig. 2.9: Classification of selection and manipulation techniques by metaphor [133]

Another classification is proposed by [19], in which selection techniques are categorized by subtasks and classified according to them. As can be seen in Figure 2.8, a selection technique must provide the indication of object, a

command to confirm the selection, and a visual aural or haptic feedback to guide users during a selection task.

Initially, researchers have been interested in virtual hand pointing techniques by mapping virtual tasks with real tasks. Although, this mapping seems to provide a natural interaction, afterwards, it has been shown that limiting the physical movements of limbs in real life provides high throughputs such as using go-go techniques by extending virtual hand to enable reach the object beyond arm-reach[132], or using virtual pointing techniques such as laser pointer metaphor (i.e., raycasting [110]). Raycasting is one of the powerful and popular pointing techniques for 3D selection tasks in virtual environments. There have been many previous studies showed that raycasting virtual pointing techniques result on better selection effectiveness [21]. Unlike classical virtual hand techniques, virtual pointing techniques enable users to both select and manipulate objects beyond arm-reach with less physical movement.

Raycasting

For selection in immersive 3D environments, Liang and Green [92] implemented a ray which is directed from the user's hand, so the user has control over the origin and orientation of the ray, which is called as raycasting technique. Raycasting uses a virtual hand (wand) position and the direction of the virtual ray to define the pointing direction. Raycasting implementations vary according to how the ray is controlled, the ray shape, and the way to disambiguate among intersecting multiple objects to decide an actual selection.

Mine et al. [111] controlled the ray by using both hands: one hand controls the position of raycasting, while other hand specifies the direction of the ray. The problem regarding of two-handed pointing is that it requires both hand tracking. However, two-handed interaction separates Degree-of-freedom (DOF), and this separations has been shown that it improves performance compared DOF integration for selection and manipulation tasks [102].

However, raycasting performance decreases when a precise selection is necessary, typically, that of small or distant objects [135] [19]. The reason is that hand trembling and input jitters are amplified by increasing distance, requiring high angular accuracy.

Although raycasting is powerful technique to select objects within hand reach and distant objects; it presents difficulties to select small or distant objects. Using "spotlight selection" [92], a cone with its apex at the user's hand, improves the selection of smaller and remote targets, while increasing the need

to disambiguate among potential targets.

Disambiguation mechanism distinguishes which object on the ray's depth is of interest. Without any disambiguation mechanisms, a system selects the closest object to the ray origin on the ray's path. Therefore, disambiguation approach can be frustrating when the target object is not the first one the ray crosses. In addition to this, disambiguation mechanisms can be classified in three types: manual, heuristic and behavioral [8]. In manual disambiguation approach, the user performs an additional selection between the set of highlighted objects, laying out the objects differently, such as with a linear or a flower menu [46], or by allowing the user to iterate among the selectable objects [160]. Using a 3D cursor along the ray is another alternative in which the object closest to the cursor is selected [46]. Heuristic disambiguation approaches "guess" the target that the user wants to select through a predefined algorithm. Finally, behavioral disambiguation considers the user's actions in a given time window and ranks objects before the selection event is triggered [49][160][125]. Despite these limitations, experimental results showed that raycasting outperforms other 3D interaction techniques in 3D selection tasks [21].

3D Cursor

3D cursor, a cursor navigates in X, Y, and Z axes, is a more direct way than raycasting [61] [132] [110]. Mine [110] suggested that a direct mapping from user's hand to a 3D cursor should be used for arm-reach interactions, while for selection at a distance, raycasting approaches such as "laser beams or spotlights" could be used. He also suggested to users removing the virtual cursor if the objects for selections are out of reach.

Go-Go Technique

Go-go selection technique is an enhanced 3D cursor technique and designated for objects at a distance [132]. The go-go technique allows to select objects both within arm's reach and remote. The movement of the virtual cursor is mapped to movement of the user's hand. However, the Go-Go technique uses a non-linear mapping by simply extending hand further than two-third of arm length to reach remote objects.

2.6 Collaborative Displays

Multi-user tabletop displays are collaborative digital tabletops that enable collaborative activities among a group of users, such as group design, sharing, and planning. Collaborative 3D displays allow a group of users to create, modify, and visualize 3D models, as well as provide remote collaboration to present modifications, and receive feedback from collaborators regarding changes through user interface [120] [128].

Interface issues are one of the most important concern in collaborative displays. Early systems address this challenge by managing multiple input devices or providing equal user interfaces for each user as well as sharing screen between collaborators, providing feedback to one without disturbing others. Greenberg et al. [43] categorized the issues as "concurrency control problems" [43]. Bier and Freeman [15] provided separate keyboard and mouse each for two collaborators to manipulate a shared content in a single computer. After that, various types of input devices has been used such as hand-held mobile computers (PDA) [117]. There are some solutions to avoid conflicting interaction such as using locking mechanisms, following social protocols, or coordination policies [113].

2.6.1 Co-located and Remote Collaboration

Teams in workplace are expected to collaborate across different physical locations. Therefore, a collaboration can be realized in two ways according to physical location: Co-located collaboration and remote collaboration (See Figure 2.10).

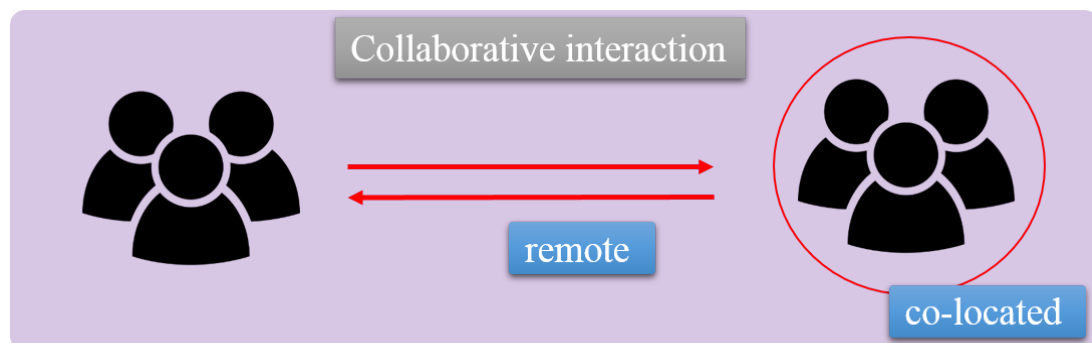


Fig. 2.10: Co-located and remote collaborations

When multiple users participate in tabletop activities, they often interact with the content on the table surface simultaneously [169]. Generally users

are forced to share the same input device when working together at a single computer. This limitation still exists in large-screen displays, which sometimes interfering with users' actions during collaboration [159]. Like large direct-input wall interfaces, tabletop interfaces require careful placement of user interface controls to avoid exhausting reaching. Similarly, direct input devices often lack of button to trigger actions yet offer greater scope for gestural and bimanual techniques. Tabletop interfaces introduce further problems because people interact around the table from different sides, whereas conventional user interfaces use a single orientation for the presentation of text and placement of controls. As Scott [148] observed, tabletop interfaces accordingly present both challenges and opportunities for new interaction design.

A number of research projects have used large horizontal interactive surfaces to present a shared content for remote collaboration, because tabletop approach similarly might provide benefits to shared workspaces for remote collaboration. An important issue regarding remote collaboration is telepresence, which helps to remote members feeling of existence of shared environment. BeThere [154] is an example for remote collaboration, which enables remote users to perform virtual interactions, such as pointing, in a local user's physical environment by utilizing a depth sensors to track position of a user's fingers, as well as to capture the 3D shape of objects in front of the sensor. In another study, a remote telepresence robot has been proposed to enable remote workers to live and work with local coworkers as if they were physically there [85]. Leithinger et al. [89] proposed Physical Telepresence which based on shared workspaces with the ability to capture and remotely render the shapes of people and objects. They investigated how the representation of user's hands can be reformed to enlarge their capabilities for tele-operation.

2.7 Summary of Background Literature

I have first provided the taxonomy of 3D displays and working principle of some types of 3D stereoscopic displays. I have also presented various studies, and 3D interaction techniques from previous literature which relates to the exploration of interaction techniques and user interfaces for both single and multi-user displays.

3

IMU-based Raycasting for 3D Selection for a Single-User Display

3.1 Introduction

Head-mounted displays (HMDs) are one of the most common single-user type 3D displays. See-through HMDs enable augmented reality (AR) applications by blending virtual objects in mid-air with the real. Devices like EPSON's Moverio [36], Sony's SmartEyeglasses [153], and META's SpaceGlassess [104] are commercially-available HMDs that support these novel AR experiences in a mobile form-factor and at a fraction of the cost of earlier solutions. At the core of this new generation of devices are low-cost tracking technologies that allow HMDs to interpret users' motion in the real world in relation to the virtual content for the purpose of navigation and interaction. Whereas previous AR devices required external and expensive equipment to track the users' motion, newer devices use techniques such as marker-based tracking [119], SLAM [180], and IMU-based dead-reckoning [116][172]. These tracking technologies enable AR applications to take place outside the research lab, in mobile settings without the need of fixed tracking infrastructure, and at an affordable price. However, the advantages of pervasive tracking come at the cost of limiting interaction possibilities. Off-the-shelf devices often still depend on peripherals such as keyboard and mouse for user input, which reduces mobility, or hand-held touchpad controllers: input mechanisms that were created for other settings and conditions of use.

Raycasting, an interaction technique widely explored in traditional virtual reality (VR), is another alternative available for AR [8]. Extensive research has

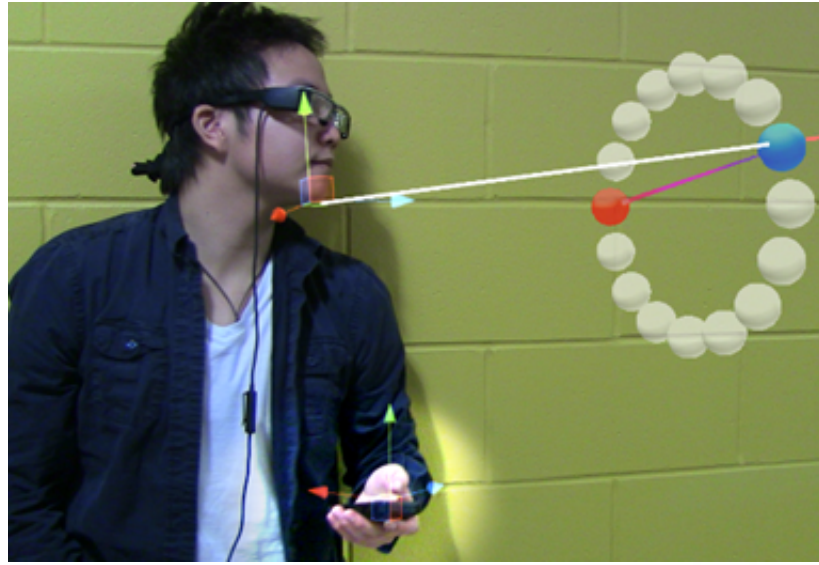


Fig. 3.1: GyroWand enables raycasting in HMDs. In this example, the ray origin is the user's chin. The ray direction is controlled using the IMU on the hand-held controller

explored the benefits of raycasting over other interaction techniques and there exist multiple design options to improve its performance [21]. Raycasting generally requires absolute tracking of the user's hand or of a hand-held controller (known as a wand), but the limited tracking capabilities of novel AR devices make it difficult to track this controller. Even when the HMD is equipped with cameras to track the environment or hand gestures, a hand-held controller is still normally outside the field-of-view (FoV) of the camera, for comfortable use. My goal is to explore how to bring raycasting to AR HMDs, even when the controller lies outside the HMD camera's FoV. This research aims to devise a way to reduce the tracking requirement from both the wand's exact position and rotation to simply its rotation. The wrist is highly dexterous [138] and with sensors readily available with HMDs ([36] [153]), I consider how best to design a raycasting interface. In principle, the IMU rotation cannot be used directly to determine the direction of the ray due to intrinsic problems such as magnetic interference and sensor drift. Moreover, the user's movement in space creates a situation in which the display contents and ray direction are often not aligned with the HMD's FoV. To address these challenges, and the central contribution of this work, I introduce a raycasting technique called GyroWand (see Figure 1). GyroWand enables raycasting in AR HMDs using IMU rotational data from a handheld controller. GyroWand introduces four fundamental design differences from traditional raycasting:

- GyroWand interprets IMU rotational data using a state machine which includes an anchor state, an active state, an out-of-sight state, and a disambiguation state (see Section 4);
- GyroWand compensates for drift/interference by taking the orientation of the hand-held controller as the initial rotation (zero) when moving from the anchor to the active state.
- GyroWand does not initiate raycasting from the user's hand, but rather from any spatial coordinate (e.g. chin, shoulder);
- GyroWand provides three novel disambiguation methods: Lock&Drag, Lock&Twist and AutoTwist. The second contribution is an experimental exploration of how best to design GyroWand for 3D content selection.

I present results of two controlled user studies revealing the most suitable design parameters: the origin of the virtual ray and the disambiguation mechanism. The final contribution is a discussion of the lessons learned for the adoption of raycasting in AR HMDs.

3.2 Related Work

Raycasting has a long tradition in 3D user interfaces for virtual reality applications [8] [21][126]. In raycasting, a virtual ray is casted from a point of origin along a given direction in the virtual environment.

When a trigger event is issued by, for example, pressing a button on a hand-held controller, the system selects one of the virtual objects crossed by the ray – typically the object closest to the ray. Benefits of raycasting include having a precise pointer, jitter control and being a natural metaphor for distant pointing. However, raycasting also presents limitations such as poor control for object manipulation (translation, rotation) and the need to disambiguate among possible targets in depth when working in dense 3D environments. Despite these limitations, experimental results show that raycasting outperforms other 3D interaction techniques in selection tasks [21]. Raycasting implementations vary in various aspects including how the ray is controlled, the ray shape, and the method to disambiguate among possible targets. Controlling the ray requires a point of origin and a direction. These two values can be provided by tracking the position and orientation of a controller or wand [46][111][126], the bare user hands [49][104], the head [170], or a mix of hands and head

[130]. For example, in Occlusion Selection the ray starts from the head and goes through the user's hand before hitting the target [130]. Another proposal is to use both hands to control the ray which adds the capacity to bend the ray by rotating the dominant hand [125]. Nonetheless, the most common approach relies on tracking a hand-held controller which can also help trigger selection events. Another important variation is the shape of the ray: it can be an actual line in 3D space [8], or it can have an aperture angle providing a cone [39]. Using a cone improves the selection of smaller targets and compensates for hand trembling and jitter, while increasing the need to disambiguate between possible targets. Finally, another variation between raycasting implementations is the disambiguation mechanism, to distinguish which object on the ray's depth is of interest. When no disambiguation mechanism is offered, the system selects the object on the ray's path which is closest to the ray origin. This approach can be frustrating when the desired object is not the first one the ray crosses. Argelaguet and Andujar [8] classify disambiguation mechanisms in three types: manual, heuristic and behavioral. Manual disambiguation allows the user to perform a secondary selection between the initial set of highlighted objects, laying out the objects differently, such as with a linear or a flower menu [46], or by allowing the user to press a button on the controller to iterate among the selectable objects [160]. Another alternative is to use a 3D cursor along the ray: the object closest to the cursor is selected [46]. Heuristic disambiguation uses a predefined algorithm to "guess" the target the user wants to select. A common approach is to select the object closest to the ray or to the central axis of the cone. A more elaborate approach uses probabilistic models to determine the selection target. Finally, behavioral disambiguation takes into account the user's actions in a given time window to rank objects before the selection event is triggered [49][160][125]. Ranks can take into account distance-to-cone axis, the time the object has been inside the cone, the number of visible pixels, etc. In this work I explore similar aspects of control, shape and disambiguation for AR HMDs. Particularly, I use IMU-data to control the ray when the location and orientation of the controller are unknown. I also propose a cone of adjustable aperture as the ray shape. Finally, I investigate different disambiguation mechanisms that rely solely on rotational data. Control, shape and disambiguation are three important aspects of a raycasting solution, although not the only ones. A more detailed account of the design and evaluation of selection techniques for 3D environments, and the design space of raycasting techniques in particular, can be found in [8].

3.3 Using IMU-data

Optical head-mounted displays for augmented reality require three key components: a computing unit, an optical element capable of showing digital content alongside the real-world, and a set of image and movement sensors. Such sensors have been the focus of considerable research aimed at providing spatial tracking without relying on external infrastructure. Research efforts include computer vision (CV) techniques like camera-based feature extraction and matching now available in commercial SDKs such as Metaio [105] or Vuforia for Smart EyeWear [183]; simultaneous location and mapping (SLAM) using depth sensors [141] and traditional low-end RGB cameras [180]; and dead-reckoning based on improved IMU data processing algorithms [172]. The main purpose of these approaches is the accurate positioning of the HMD in relation to the virtual world. Therefore, they provide limited support for the tracking of objects out of range for the HMD sensors (i.e. camera). Objects need to be within the field of view of the camera. This way, users need to position their hands or objects within the camera FoV by either raising them in front of the HMD or by looking downwards towards the object. Not only are these adjustments physically demanding (raising the hands [59]) and unconformable (looking down), but they also interfere with the user's task. Such limited tracking also impacts the way in which raycasting can be realized in AR HMDs, particularly its control mechanisms: only raycasting from the head [170] and from the head through the hand [130] seem to be available. In this research I aim to leverage sensors not located on the head piece for interacting with content. For example, the EPSON Moverio BT-200 is made of an optical HMD and a hand-held controller [36]. The controller contains the computing unit and a touchpad for interacting with the HMD applications. The controller also provides a 9-axis IMU which can capture the controller orientation. My approach is to leverage this rotational data for controlling the direction of the ray. For other HMDs without a hand-held controller (such as the announced Microsoft HoloLens [108]), using external objects to capture rotational data such as Bluetooth connected wands, rings or even smartphones can be imagined.

3.3.1 Challenges Associates with IMU data

Despite the convenience of using an available hand-controller, the rotational data gathered from IMUs has several known problems: noise, sensor drift and magnetic interference, and axis mapping. Noise relates to the quality of the

measurement. A noisy sensor can exaggerate the amount of rotation or report rotational movement where there was none. Noise handling is important because raycasting derives its naturalness from being an isomorphic interaction technique, i.e. there is a one-to-one mapping between movement of the controller and the virtual ray. Traditional noise reduction strategies such as thresholding or a control-display (CD) ratio are inconvenient because they compromise isomorphism. The ultimate solution to the noise problem is higher quality IMU electronics. For raycasting, noise means that the direction of the ray will present jitter or trembling even if there is none. Sensor drift is a related problem where the sensor might report a slow yet continuous change in rotation where there is none. Newer IMU sensors appear to successfully handle drift: the direction of gravity can be used to cancel out drift in the X and Z axis, while the compass can stabilize drift along the Y axis. Nonetheless, both strategies fail when the amount of movement is beyond or below the sensors' tolerance or when there is magnetic interference that affects the compass. Better tracking algorithms are able to reduce drift [98] [172] but not eliminate drift. For raycasting, drift means that the same controller orientation results in different ray directions as drift accumulates. Finally, the relation between the coordinate systems of the virtual content and that of the controller's IMU changes as the user interacts with the application or moves in the real world. For example, if the HMD application uses view-centric content (content that is always visible regardless of the head orientation) and the user rotates his general orientation 90 degrees on y, the resulting virtual ray moves also 90 degrees in relation to the virtual content, that is, out of view for the user. Similar mismatches occur when the user turns around to continue interacting with the virtual counterpart of a physically moving object. In both cases, the user has to accommodate the orientation of the controller in relation to the real world, instead of in relation to the virtual world. In the first example, after rotating the body 90 degrees, the user would have to maintain the controller in the same orientation in the real world as before rotating his body. This mismatch in coordinate systems leads to an unnatural mapping between the movements of the controller and the ray, which is one of the strengths of raycasting. In summary a raycasting solution using rotational data as provided by an external IMU should address the following requirements:

- **R1** – Provide mechanisms to compensate for noise-induced jitter.
- **R2**– Assume an interaction model acknowledging the continuous effect of

sensor drift in the direction of the ray.

- **R3** – Recognize the changing relation between the coordinate systems of the handheld controller and the virtual content.

Also, from the literature on raycasting I know that a raycasting solution should address the following:

- **R4** – Provide an alternative origin for the virtual ray given that the actual controller position cannot be tracked.
- **R5** – Propose disambiguation techniques that leverage the degrees of freedom available to the controller.

Section 4 presents the design of GyroWand, focusing on R1-3. To address R4 I explored various regions for originating the virtual ray which I present in Section 5. For R5 I created and evaluated three novel disambiguation mechanisms presented in Section 3.5

3.3.2 IMU-based Raycasting

I have implemented a raycasting technique, called GyroWand, for self-contained AR HMDs. Here, self-contained means the lack of external tracking of the head piece, hand or objects. GyroWand directs a ray originated at a predefined location in virtual space (see Section 4) using rotational data acquired from an IMU in the user's hand.

Dynamic Apex

To address R1, the effect of noise on jitter and selection accuracy, GyroWand uses a dual approach. First, the data is filtered with a moving window from the last five data points. I do so to reduce jitter while maintaining isomorphism between the movements of the controller and the ray. Second, I use a cone shape with a 2 degrees aperture apex (as used in [46]). Moreover, users can decrease the aperture apex to 0 degrees by twisting the controller 45 degrees inwards (pronation). Similarly, users can increase the aperture apex to 6 degrees by twisting the controller 45 degrees outwards (supination). Figure 2 shows the effect of GyroWand's dynamic apex on the apex diameter at 1 meter away from the origin. A dynamic apex leverages best-practices (2 degrees default) while allowing users to control the apex according to the level of noise produced at a

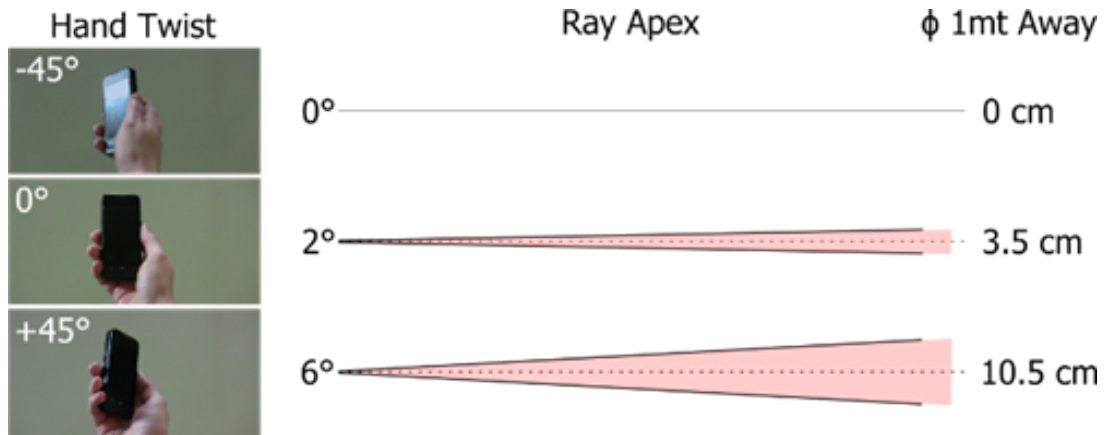


Fig. 3.2: Pronation and supination control the apex aperture

given moment. It is important to note that, regardless of the apex value, the system renders the ray as a single line.

There are other alternatives to the manual control of the aperture apex. For example, the system could monitor the noise level and automatically adjust the apex to the noise, providing a wider apex when the noise level is high and a narrow one when the noise level is low. Another alternative would increase the apex aperture according to the phase of the selection task. In the ballistic phase, when the user is moving from one target to another with large movements and thus generating more noise, the apex can be wider. In the corrective phase, when the user is refining the selection with small movements and therefore generating less noise, the apex can be narrower.

A State-Machine for Raycasting

GyroWand addresses R2 (drift) and R3 (coordinate mapping) using the state machine presented in Figure 3.3. GyroWand is always in one of four states: Anchor, Active, Out-of-Sight, or Disambiguating. GyroWand transitions between states manually by rotating or touching the controller, or automatically via timeouts. The initial state is Anchor, and it is set during the initialization process. In the anchor state GyroWand points its ray to a pre-defined coordinate in space relative to the HMD called the anchor point. When users move or rotate their heads in virtual space the anchor point remains at the same position relative to the HMD and therefore the ray, which points to the anchor point, seems static to the user. Rotational data has no effect on the ray direction. The ray is presented in a dark red color, indicating to the user that it is in the anchor state and therefore disabled for interaction.

Tapping on the controller's touchpad activates the GyroWand, transitioning

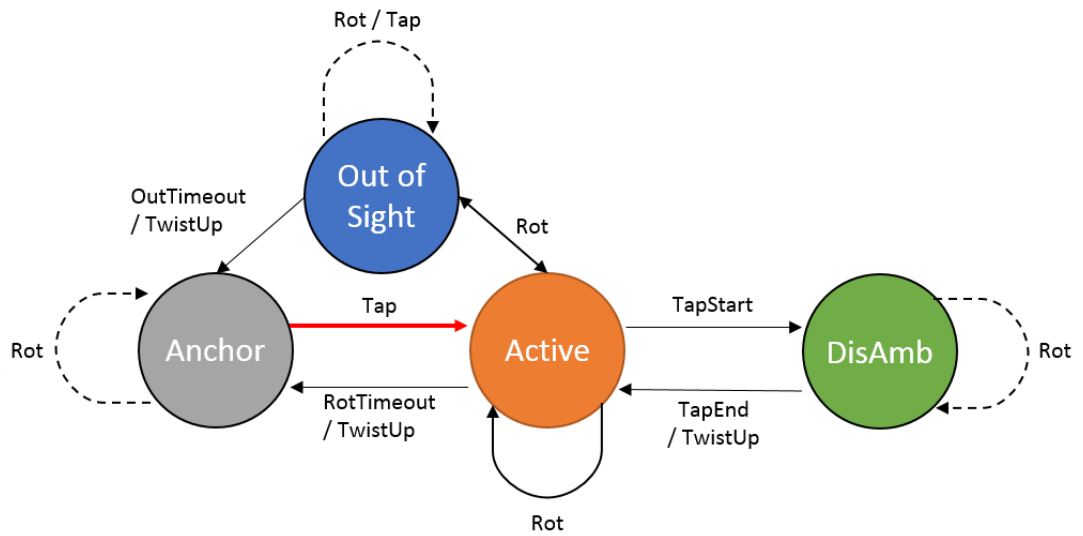


Fig. 3.3: The state machine for IMU-based raycasting has four states: anchor, active, out-of-sight, and disambiguating. The transition from Anchor to Active sets the current controllers orientation as the baseline for the GyroWand movement.

from the Anchor state to the Active state. During this transition, the GyroWand captures the orientation of the controller and uses it as the initial or baseline rotation; i.e., the rotation at which the active ray points at the anchoring point. All rotational movement of the controller is now applied to the ray direction. The GyroWand notifies all the virtual objects it crosses. A quick tap event ($< 150\text{ms}$) selects the hovered object closest to the ray. Users can transition back from the active to the anchor state in two ways: a rotation timeout or a TwistUp gesture. The rotation timeout, currently set at 5 seconds, transitions the state back to anchor when the total rotation on all axes is smaller than a given threshold. This timeout serves as an implicit interaction to disable the ray when the user places the controller on a surface (e.g. a table) in order to work with real world objects. The user can also issue a TwistUp gesture (akin to pulling a fishing rod) which is implemented as a rotation on the x axis for more than 45 degrees in less than 150 milliseconds. In the presence of drift (R2) the ray slowly changes direction even as the physical controller is still, leading eventually to the ray leaving the display area. Naturally, a user compensates drift by rotating the controller in the opposite direction. However, when drift is large, users can reset the initial rotation by moving back into the anchor state (TwistUp gestures) and back again into the active state (tapping the touchpad). Similarly, moving between the anchor and active states, thus setting the hand-held controller initial or baseline rotation, helps users to manually deal with conflicting coordinate systems (R3). If the controller was motionless while users moved around in physical space,

the next time the GyroWand is activated the ray will point to the anchor point within the current view. If the user is moving around while the GyroWand is active, he can reset the baseline rotation by going back into anchor state and back again to active. The GyroWand's ray can at times be out of the user's sight, that is, outside the small FoV of the HMD. For example, the EPSON Moverio BT-200 has a diagonal FoV of 23 degrees (11.2 degrees vertical). Accumulated drift or user movement in the real world could end up with an out-of-sight ray. User interactions with the hand-held controller or an accommodation of the hand to reduce fatigue can also lead the controller to an orientation that takes the ray of out sight. A ray out of the user's FoV can be challenging to recover because the user has no indication of how the hand-held controller must be rotated to bring the ray back into view. GyroWand enters into the Out-of-Sight state automatically when it judges the ray is out of the HMD's FoV. In the Out-of-Sight state touch input is ignored to avoid triggering selection events on virtual objects out of the user's view. The user can bring the GyroWand back into active state by correcting the controller orientation. The user can also issue a TwistUp gesture to anchor the ray, and then go back to active state. If the user does not correct the ray direction and no TwistUp gesture is issued within a given timeout (currently 5 seconds), GyroWand transitions into the anchor state automatically. This transition is aimed at reducing the time a user spends "looking" for the ray. The Disambiguation state is where users refine the actual target they want to select in depth. The GyroWand goes into the Disambiguation state only when a tap is started from the active state and the ray crosses more than one virtual object. Taps in the anchor and out-of-sight states are ignored. In the Disambiguation state the ray is locked on the position and orientation where the tap was initiated. The default selection target changes from the object closest to the ray to the one closest to the origin. The user can then use any disambiguation method to iterate over the highlighted objects and specify a new selection target. Disambiguation ends when the user releases the finger from the touchpad, issuing the selection event on the current selection target and returning to the active state. Section 6 presents three disambiguation mechanisms which the user controls while in the Disambiguation state.

3.4 On-body Raycasting Origin

Given that GyroWand uses only rotational data, it does not know the spatial location of the controller in the real-world. Therefore, a suitable body location

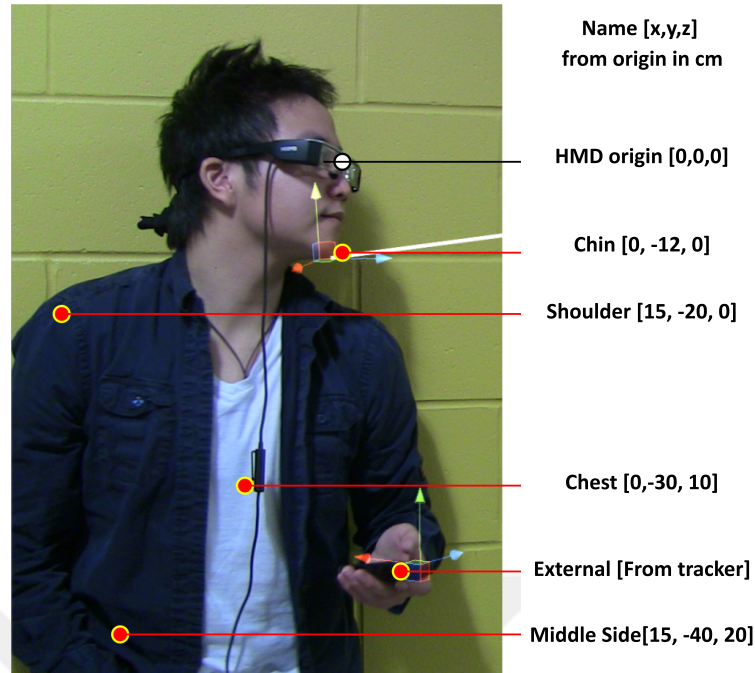


Fig. 3.4: Four possible virtual positions that explored for the GyroWand's ray origin: chin, chest, shoulder, and middle side. External represents the actual controller location measured with an optical tracker and used as baseline.

for the origin of the ray must be determined. Using the eyes' center as the origin can be an initial idea. However the visual representation of such a ray is confusing at best: the ray is shown too close to the users' eyes resulting in poor 3D visual effect (often visible in one eye only), occupies considerable display real estate, and causes eye strain. Therefore I chose to explore different body locations (see Figure 4) inspired partly by observations of the way in which people hold and carry the HMD controller, and by my design assumptions. The first ray origin is Middle Side (MS) and is located at (15,-40, 20), that is 15 cm to the user's dominant side, 40 cm below the virtual camera and 20 cm in front of the user. I chose this location because it most closely resembles a user's hand position when holding the HMD controller while standing. The second ray origin is the Chest (CT) and is located at (0, -30, 10). The Chest captures the hand position of users who hold the HMD controller in front of their chest, similar to the way smartphones are held when typing. The third ray origin is the Shoulder (SH) and is located at (15, -20, 0). I selected the shoulder to compensate for differences in the physical dimensions, mainly height, between users. The final ray origin is the Chin (CN) and is located at (0, -12, 0). I study the chin because the effect of the distance to target on the ray direction is less severe than that of other locations. For example, when content is very close to

the user, the controller would have to be held almost vertical when coming from the chest, and almost lateral when coming from the shoulder. These orientations cause a mismatch between the hand movement and the capacity to acquire a target. A ray coming from the chin would almost always move forward and within the HMD FoV. Nonetheless, these differences minimize rapidly as the distance to the target increases.

3.4.1 Study 1: Raycast Origin

In this experiment I investigated user efficiency and physical exertion for the different ray origins in 3D selection tasks in different reference frames. I compared the virtual origins to the origin of the actual controller location.

Apparatus

Head registration was provided with an external optical tracking system OptiTrack with 6 cameras covering a cube of dimensions 1 (wide) x 1 (depth) x 2 (height) meters. The middle of the cube is located at 1 meter height from the floor. The system tracked the location and orientation of the smart glasses at 60 FPS and transmitted the data to the smart glasses via UDP over WiFi at a 40 FPS (a 30 percent reduction due to networking overhead). The participants stood in the center of the tracking cube. For the smart glasses I used the EPSON Moverio BT-200 with a 23 degree diagonal FoV, a 960x540 pixel display resolution, and a focal depth of 5 meters. I created stereoscopic 3D graphics using the side-by-side approach at 50 FPS. Touch input was received through the touchpad of the Moverio BT-200.

Subjects

15 participants (5 female) volunteered, ages 18-33 (mean 26), all right handed. Four participants had previous experience with head-mounted displays, three had experience with virtual reality, and none of them had experience with raycasting.

Task

For each session participants had to sequentially select 16 targets in a circular layout on the same plane. The system shows all targets in grey, and the target currently crossed by the ray in light green. The next selection target is shown in red. When the selection target is crossed by the ray its color changes to a

brighter red. Selection was triggered upon releasing the finger from the touch pad. Upon selection the system shows the target in magenta, and highlights in red the object in front of the current one as the new target. A line colored red-to-blue also indicates the location of the new target, guiding the participant. This is an important consideration for situations when the new selection target is located out of the field-of-view (see Figure 3.1). The session ended when the participant selected all 16 targets. A selection error was marked when selection was triggered on another object or the void.

Design

Independent variables were ray origin and reference frame. I used a 5x3 within-subject design to compare user performance. I considered the four ray origins (Chin, Chest, Shoulder and Middle Side) and the real controller location as baseline (tracked with an optical tracker). I located 3D content at the same distance from the HMD (50 centimeters away) but on different reference frames. I am interested in the effects of reference frame rather than distance for two reasons. First, it can be safely assumed that as the distance increases the impact of the origin is negligible because their differences are less than the distance to the target. On the other hand, when objects are close to the user, the origin of the ray can greatly impact the rotational movement needed to reach a target. Second, for content close to the user the reference frame makes a difference as content reacts differently to user movement. I evaluated three reference frames: view, body and world [35]. With the View reference frame content remains fixed in the field-of-view regardless of any head movement. On the Body reference frame content is always at the same location in relation to the body. For example, content positioned on the right of the body will always be visible when the user turns his head to the right, but not when he looks forward. Finally, content on a world reference frame remains fixed in space irrespective of the user location or gaze direction; the targets were located at the center of the tracking cube and 1.5 meters high, participants stood 50 cm away. Random variables were target size and target distance (diameter of the circle). I considered target sizes of 1 and 2 centimeters of diameter. I considered target distances of 7 and 9 centimeters. I chose these seemingly small dimensions because of their large size in the HMD particularly for the view reference frame in which all targets must fit within the display (see Figure 3.5). Participants were asked to hold the Moverio BT-200 controller in their dominant hand. All rotational information was calculated using the internal IMU of the Moverio BT-200 controller. For the

real controller location and orientation (controller condition), I used optical tracking, so that it could be accurately attached to the user hand. At the start of each condition, the experimenter asked the participant to hold the controller in an initial and comfortable position; this position was set as the rotation baseline (the ray was horizontal and forward from the active source). The experimenters demonstrated the task and participants were allowed to train for 10 minutes. A minimum of 20 selections with each ray location were required to complete the training session. The trials were counter-balanced using a latin-square design on the 15 conditions. Target size and distance appeared in random order. With a total of $5 \times 3 = 15$ conditions per participant and $16 \times 2 \times 2 = 64$ selections per condition, I registered $15 \times 64 = 960$ selections per participant or 14400 selections in total. All participants completed the experiment in one session lasting approximately 60 minutes.

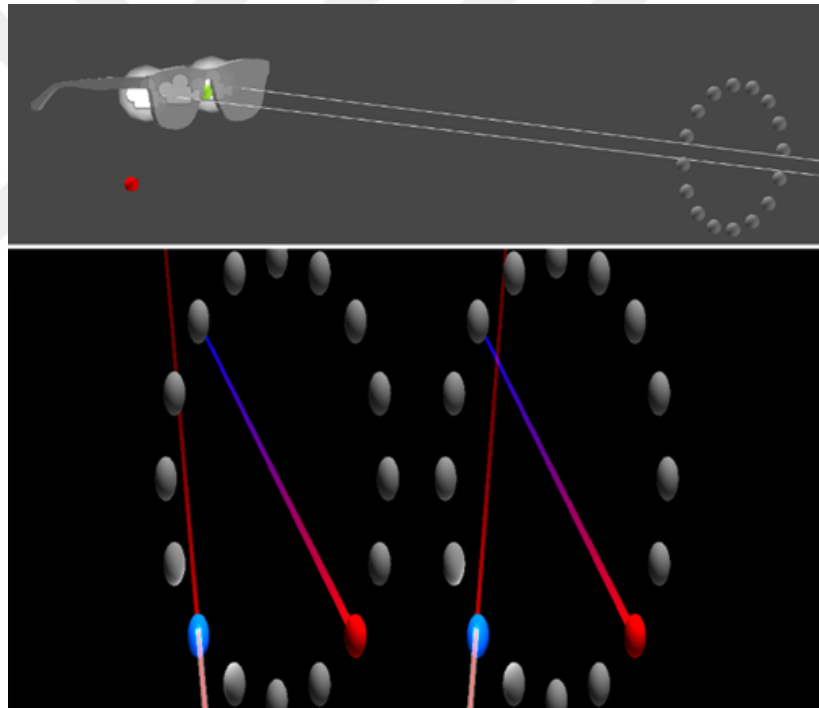


Fig. 3.5: Targets of 1cm width and 9cm separation. Top: Targets are located 50 in front of the HMD. Bottom: Stereoscopic rendering of the user view

Measures

For each trial task completion time and selection errors were recorded. For each condition Borg RPE ratings of physical effort was collected.

3.4.2 Results

I removed outliers after 3 standard deviations of completion time. Figure 6 presents the results I analyzed using analysis of variance (ANOVA) and post-hoc tests with Bonferroni corrections. Completion time – Results showed a main effect for origin ($F_{4,56} = 5.080, p = 0.001, \mu^2_{\text{partial}} = 0.266$) and reference frame ($F_{2,28} = 5.448, p = 0.01, \mu^2_{\text{partial}} = 0.280$). There were no significant interaction effects for origin $\times$ reference frame ($p = 0.385$). Post-hoc tests showed significant differences between all origins ($p < 0.001$) except, between Chest and Shoulder ($p = 0.250$) and between Middle Side and real controller origin ($p = 1.0$). Chin and Shoulder were the origins with the fastest completion time at 1.737 sec ($stdev = 0.663$) and 1.798 sec ($stdev = 0.698$) respectively. Post-hoc tests showed significant differences between all reference frames ($p < 0.001$), except between Body and View ($p = 0.096$). Participants were fastest in View (1.826sec, $stdev = 0.708$) followed by Body (1.854 sec, $stdev = 0.713$) and World (1.921sec, $stdev = 0.734$). Selection errors – Results showed a main effect for origin ($F_{4,56} = 9.293, p = 0.001, \mu^2_{\text{partial}} = 0.398$) and reference frame ($F_{2,28} = 7.307, p = 0.003, \mu^2_{\text{partial}} = 0.342$). There were no significant interaction effect for origin $\times$ reference frame ($p = 0.057$). Post-hoc tests showed significant differences between all origins ($p < 0.05$), except between Chest and real controller ($p = 1.0$), between Chest and Shoulder ($p = 1.0$), between Chin and Shoulder ($p = 0.096$), and between Shoulder and real controller ($p = 0.504$). Chin and Shoulder were the origins with the lowest errors rate across all reference frames at 0.17 ($stdev = 0.455$) and 0.21 ($stdev = 0.517$) respectively. Post-hoc tests showed significant differences between all reference frames ($p < 0.05$). Participants recorded the least number of selection errors in Body (0.21, $stdev = 0.499$) followed by View (0.23 sec, $stdev = 0.53$) and World (0.26, $stdev = 0.57$). Borg RPE - Results did not show a main effect for origin ($p = 0.091$) or reference frame ($p = 0.06$).

3.4.3 Discussion

These results show that an origin near to the user's chin seems to be the best choice, regardless of the reference frame. I observed significant differences for the View and Body reference frames, and an equivalence to other origins relatively close to the HMD (Chest and Shoulder) on the World reference frame. This finding supports my assumption that when content is close to the user, the relative distances between the origins are an important factor. The observed

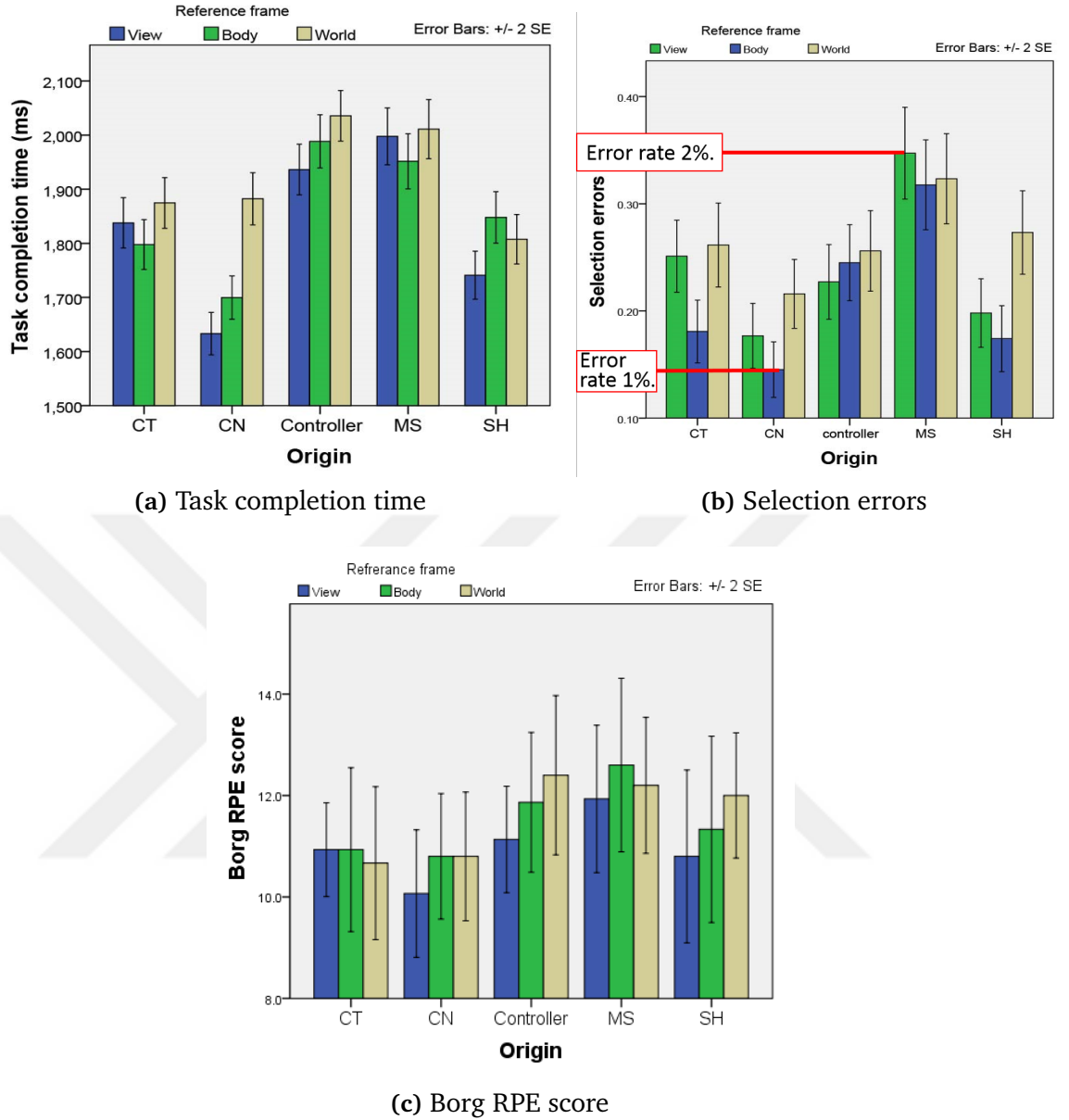


Fig. 3.6: Quantitative results comparing different ray origins. The baseline condition of the actual controller location, equivalent to traditional raycasting

advantage of Chin over the other origins was not only in terms of completion time, but a low error rate was also observed; which tells about the controllability of the GyroWand on a particular origin. Moreover, participants did not report any extra fatigue on the Chin, compared to other origins. From these results I recommend GyroWand to employ a location near the user's Chin as the origin. An interesting observation is that raycasting from the actual controller location shows some of the worst performance metrics. It shows, together with Middle Side which tries to mimic it, the slowest completion time and higher error rate.

This means that users actually struggle with selecting content within arm's reach with a ray coming from their hands, even when they could accommodate the controller in a more convenient position. This is an important result which could have implications on the use of raycasting in other scenarios such as virtual environments. Common approaches to disambiguating among potential targets when using raycasting include marking menus and the movement along the axis of the ray [46]. While the use of marking menus such as the Flower Ray [8] and similar approaches that spread out conflicting targets is an interesting direction, I am interested in approaches that do not require complementary user-interfaces. Moreover, moving the controller along the axis of the ray is not an option because my design goal is not to depend on external tracking mechanisms, and IMU-based estimations of spatial movement is still unreliable for the consumer-level IMUs included in off-the-shelf devices. Therefore, I focus my exploration on techniques that use IMU-based rotational movement and touch input.

3.5 IMU-based Disambiguation

Common approaches to disambiguating among potential targets when using raycasting include marking menus and the movement along the axis of the ray [46]. While the use of marking menus such as the Flower Ray [46] and similar approaches that spread out conflicting targets is an interesting direction, I am interested in approaches that do not require complementary user-interfaces. Moreover, moving the controller along the axis of the ray is not an option because my design goal is not to depend on external tracking mechanisms, and IMU-based estimations of spatial movement is still unreliable for the consumer-level IMUs included in off-the-shelf devices. Therefore I focus an exploration on techniques that use IMU-based rotational movement and touch input.

3.5.1 Lock and Drag

The first technique is Lock and Drag (LD) and is shown in Figure 3.7. In LD the user directs the ray toward a set of objects including the object to select. Highlighted objects are shown in blue (light and dark), not highlighted objects are shown in gray. Putting the finger on the touchpad transitions GyroWand into the Disambiguation state: locking the ray and painting it green. As long as the finger presses the touchpad the ray position and orientation remain unchanged.

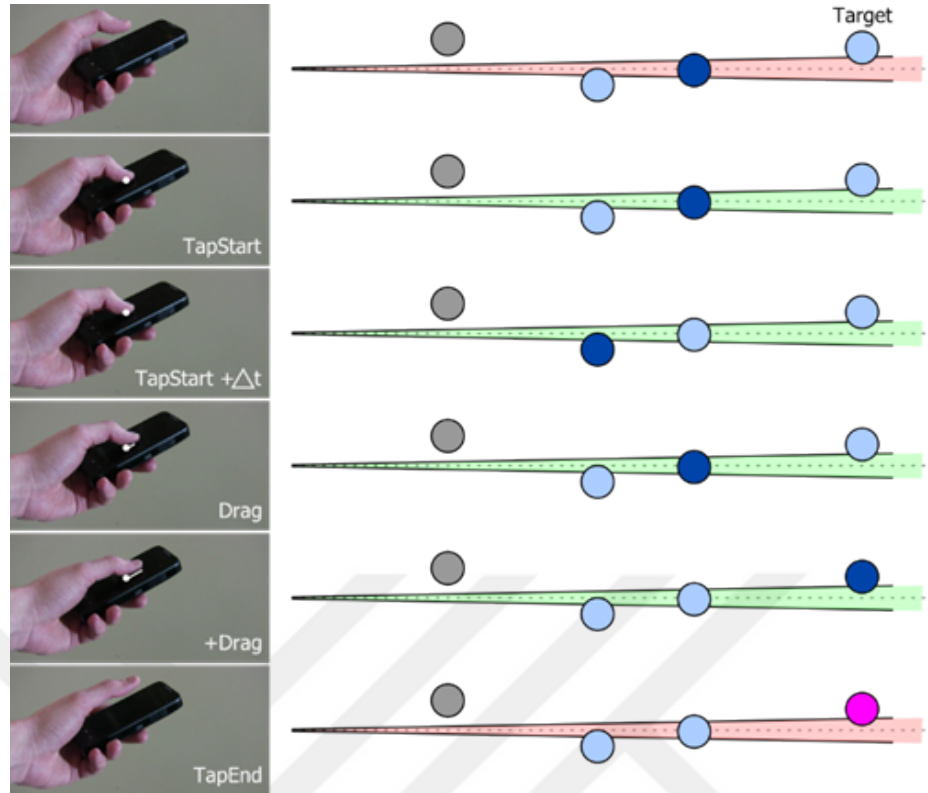


Fig. 3.7: Lock&Drag. The finger down starts the disambiguation. Dragging the finger up and down refines the selection. Lift the finger to confirm the selection event. Objects not hovered in gray, objects hovered in light blue, selection candidate in dark blue, and selected object in magenta. Active ray is pink, disambiguation ray is green.

Initially, the object closest to the ray is the selection candidate, and colored dark blue. If the user does not remove the finger within a given ($\Delta t < 150ms$), the object closest to the ray origin becomes the selection candidate. The user drags the finger in any direction away from the initial point of contact and after a set threshold (50px) the next object away from the origin becomes the selection candidate and turns dark blue, and the previous selection candidate turn light blue. Therefore, to select the first object on the ray path a simple tap + Δt is enough. Drag the finger one time the threshold to select the second object, two times the threshold to select the third, and so on. Dragging the finger back and closer to the point of contact moves the selection candidate back to the previous object. Selection is triggered when the user removes the finger from the touch pad (magenta in Figure 3.7). Locking can be a source of errors in itself. For example, the finger movement necessary to activate locking is often accompanied by hand rotation and sometimes full body movement. This small spatial and rotational displacement might cause the ray to lock at

a different location and orientation by the time the lock is activated. This phenomenon is known as the Heisenberg effect [3] and has an equivalent manifestation when the user lifts the finger to confirm the selection. This implementation corrects for Heisenberg effects by correcting the rotation and position of the ray to their respective values 50 milliseconds before locking the ray and triggering the selection event (I arrived to this number iteratively, further research is necessary for a more finely defined value). Nonetheless, even after correcting for Heisenberg effects, users might still activate the ray at the wrong position because of their manipulation of the controller or normal operational or perceptual errors. I minimized the possibility of unintentional locking by limiting the locking operation to situations in which the ray is actually intercepting an object and the intersection point with such object is inside the users' field of view. Users can unlock the ray (out of Disambiguation and into the Active state) by issuing a TwistUp gesture (rotation on X at 130 degrees/second). Finally, device has been utilized in this research (Moverio BT-200) provides touchpad data as a mouse cursor, resulting in several challenges to take into account for the LD disambiguation technique. First, the defined threshold for switching the selection candidate is defined in display units (50px) instead of actual physical units of movement and thus subject to the Control-Display ratio provided by the manufacturer. This means that faster finger movements result in larger pixel movements, even if the actual finger displacement was shorter than a larger slower movement – making it harder for novel users to understand the functioning of the technique. I explain this issue to the experiment participants and demonstrated it during training. Another consequence of obtaining the touchpad data as a cursor is that its movement stops upon reaching the edges of the display, which means that if the user had been dragging his finger forward, the user might perceive the disambiguation as not-functioning when the cursor reaches the display edge. Therefore GyroWand provides an arrow on the display suggesting a finger dragging direction always away from the edges.

3.5.2 Lock and Twist

The second disambiguation technique is Lock and Twist (LT) and is shown in Figure 3.8. This technique leverages the extra degree of freedom in wrist movement that is not normally used in my implementation of raycasting operations. This implementation uses flexion rotation along the X axis, and ulnar and radial deviation for rotation along the Y axis. LT leverages pronation and supination (a twist movement of the wrist) as an available degree of freedom. Similar to the

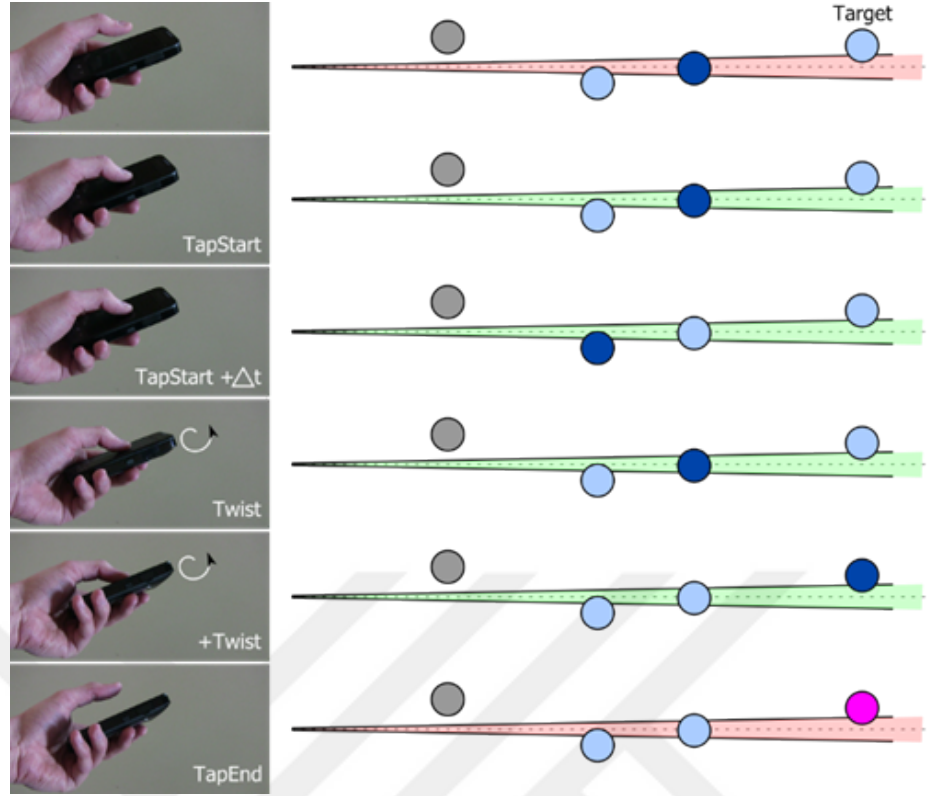


Fig. 3.8: Lock&Twist. The finger down starts the disambiguation. Supination refines the selection. Lift the finger to confirm the selection event.

previous technique, the user points the ray toward a set of objects including the selection target and moves into the Disambiguation state by placing the finger on the touchpad. Upon starting disambiguation, locking the ray, and waiting for a ($\Delta t < 150$ ms), the first object to encounter the ray is the selection candidate. The user changes the selection candidate by turning the wrist outwards (supination) by a given threshold. Based on [21] implemented a linear segmentation of the available rotational space at 4 spaces (60 degrees available for supination divided by a 15 degrees threshold). The selection event is triggered when the user lifts the finger from the touchpad. Similar to LD, Heisenberg effect is handled by correcting location and orientation at 50 milliseconds before the locking of the ray and the selection event.

3.5.3 Auto-lock and Twist

The final disambiguation technique is Autolock and Twist (AT), or simply AutoTwist and it is presented in Figure 3.9. The difference between AT and LT is the way in which the GyroWand enters the Disambiguation state. In AT the user points the ray toward the set of objects including the selection target and

issues a quick supination movement. After iterative testing the settled for a rotational change in the z axis of 7.5 degrees within 150 milliseconds (50 deg/sec). Once the trigger twist is detected, the system corrects the ray's orientation and position to those at the start of the movement. The system also corrects for Heisenberg effect at 100 ms before the final selection event.

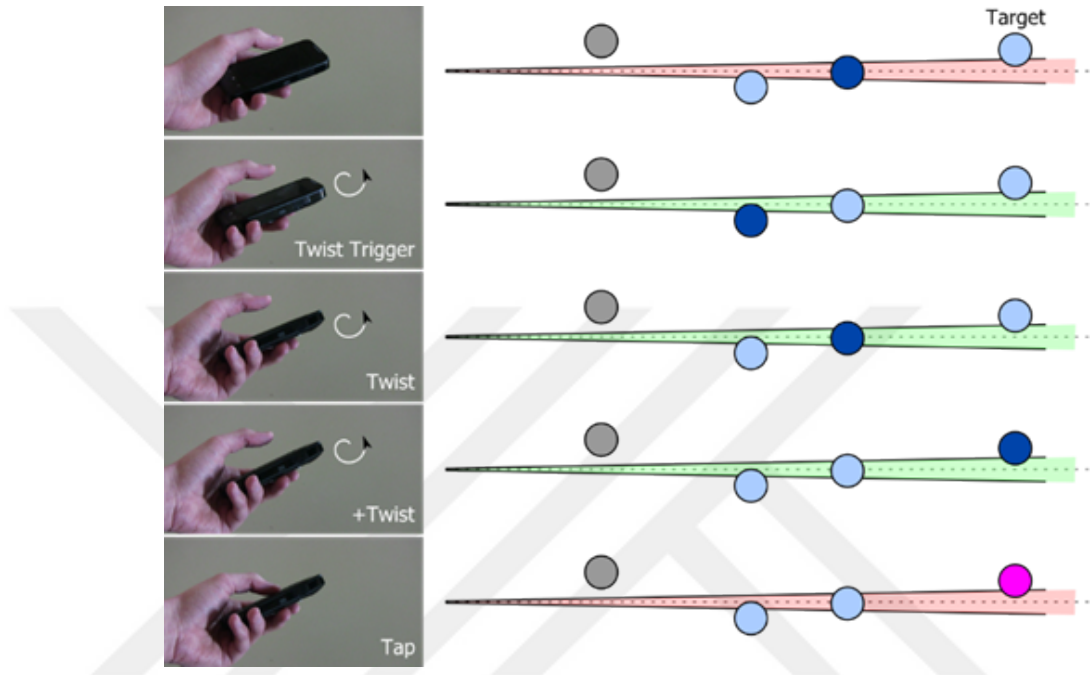


Fig. 3.9: Autolock & Twist. A quick twist starts the disambiguation. Supination refines the selection. A tap confirms the selection.

3.5.4 Study 2: Selection Disambiguation

In this experiment, the efficiency of the designed raycasting disambiguation techniques in a 3D selection task was investigated.

Apparatus

Similar to experiment 1, head registration for the Moverio BT-200 was provided by an external optical tracking system. In this case, a Vicon Tracking system with 8 cameras covering a cube of dimensions 2 (wide) x 2 (depth) x 2 (height) meters was used. The middle of the cube was at 1 meter height from the floor. The system tracked the location and orientation of the smart glasses at 60 FPS and transmitted the data to the smart glasses via UDP over WiFi at 40 FPS (a 30% reduction due to networking overhead). The participants stood in the center of the tracking cube.

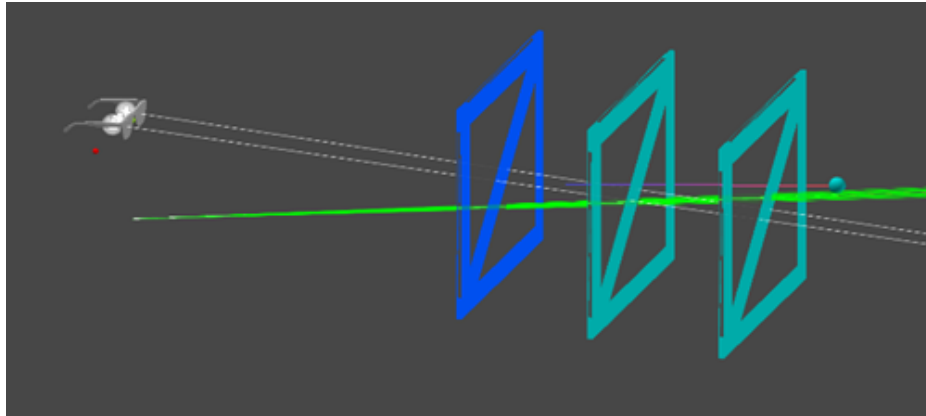


Fig. 3.10: Three obstacle walls and one target.

Subjects

12 participants (1 female) volunteered, ages 18-33 (mean 26), all right handed. Four participants had previous experience with head-mounted displays, three had experience with virtual reality, and none had experience with raycasting.

Task

For each condition participants had to sequentially select 45 targets located behind 1, 2 or 3 wall obstacles (see Figure 3.10). In each selection, the participant had to disambiguate selection between the obstacles and the selection target. In the example shown in Figure 3.10 the target is located behind the third obstacle wall. All crossed objects are shown in light blue, the selection candidate is displayed in dark blue. To select the target (sphere) the participant has to disambiguate three times, skipping the three obstacle walls, before selecting the target. For all disambiguation techniques and the baseline (PC – point cursor, see below) selection is triggered upon releasing the finger from the touch pad. Upon selection the system highlights the target in magenta, and moves it to a new location in space. A line colored red-to-blue indicates the displacement to the new location, guiding the participant. This is an important consideration for situations when the selection target is relocated out of the current field of view. The task finished when the participant selected 45 targets. All targets were randomly positioned. A selection error was marked when selection was triggered on an object different than the current target.

Design

Disambiguation technique was the independent variable. I used a within-subject design to compare user performance with each technique. I considered the three disambiguation techniques (LD, LT, and AT) and Point Cursor (PC) selection as a baseline (similar to the approach used in [8]). The number of obstacles was a random variable with equal amounts of selections. I leveraged the results of experiment 1 and used the “chin” as origin. Participants were asked to hold the Moverio BT-200 controller in their dominant hand. All rotational information was calculated using the internal IMU of the Moverio BT-200 controller. For the point cursor the optical tracking system was used to determine the Moverio BT-200 controller location and orientation, so that its source was effectively located at the participant hand (this was done to provide a reliable baseline). The point cursor was located at a fixed length of 50 cm from the controller location and in the ray direction. For all disambiguation techniques except PC participants stood 50 cm in front of the first obstacle wall. For PC participants could walk freely in the space to be close enough to be able to select the target. At the start of each condition, the experimenter asked the participant to hold the controller in an initial and comfortable position; this position was set as the rotation baseline. The experimenter demonstrated the selection task with each disambiguation technique and participants were allowed to train for 10 minutes. A minimum of 20 selections with each technique were required to complete the training session. The trials were counter-balanced using a latin-square design on the 4 disambiguation mechanism. With a total of 4 conditions per participant and 45 selections per condition, $4 \times 45 = 180$ selections per participant or 2160 selections in total was registered. All participants completed the experiment in one session lasting approximately 30 minutes.

Measures

For each selection trial task completion time, disambiguation time (calculated as the time to complete the task minus last time of arrival), and selection errors were recorded. For each condition Borg RPE ratings of physical exertion also was collected.

3.5.5 Results

outliers were removed after 3 standard deviations of completion time. Figure 3.11 presents the results analyzed using analysis of variance (ANOVA) and post-

hoc tests with Bonferroni corrections. Completion time – Results showed a main effect for disambiguation mechanism ($F_{3,33} = 13.431, p < 0.001, \mu^2_{partial} = 0.550$) and obstacles ($F_{2,22} = 10.449, p = 0.001, \mu^2_{partial} = 0.487$). Results also showed an interaction effect between disambiguation mechanism $\times$ obstacles ($F_{6,66} = 3.225, p < 0.01, \mu^2_{partial} = 0.225$). Post-hoc tests showed significant differences between all mechanisms ($p < 0.01$). PointCursor was fastest at 4.044 sec (stdev = 1.931), followed by Lock&Twist at 4.851 sec (stdev = 1.963), Lock&Drag at 5.162 sec (stdev = 2.222) and AutoTwist at 5.691 sec (stdev = 2.432). Post-hoc tests showed significant differences between all number of obstacles ($p < 0.001$) except between 1 and 2 obstacles ($p = 0.243$). Participants selected targets with 1 obstacle in 4.864 sec (stdev = 2.127), against 2 in 4.822 sec (stdev = 2.194) and against 3 in 5.267 sec (stdev = 2.3). Disambiguation time – Results showed a main effect for disambiguation mechanisms ($F_{3,33} = 152.827, p < 0.001, \mu^2_{partial} = 0.933$) and obstacles ($F_{2,22} = 19.599, p < 0.001, \mu^2_{partial} = 0.640$). Results also showed an interaction effect between disambiguation mechanism $\times$ obstacles ($F_{6,66} = 5.760, p < 0.001, \mu^2_{partial} = 0.343$). Post-hoc tests showed significant differences between all mechanisms ($p < 0.001$). AutoTwist disambiguation was fastest at 2.144 sec (stdev = 0.859), followed by Lock&Twist at 2.516 sec (stdev = 0.796) and Lock&Drag at 2.789 sec (stdev = 1.173). PointCursor did not entail a disambiguation. Post-hoc tests showed significant differences between obstacles ($p < 0.003$). Participants disambiguated 1 obstacle in 1.841 sec (stdev = 1.049), 2 in 2.048 sec (stdev = 1.215) and against 3 in 2.184 sec (stdev = 1.213). Selection errors – Results showed no main effect for disambiguation mechanism ($p = 0.204$) or obstacles ($p = 0.298$). Error rate was similar for all condition at 0.22 (stdev = 0.56). Borg RPE – A Friedman χ^2 test did not show significant differences between disambiguation mechanisms ($p = 0.06$).

3.5.6 Discussion

The experimental results show that Lock&Twist is the mechanism which is in general fastest than the others without any particular extra effect on error rate or perceived physical exertion. This advantage was present even when AutoTwist presented a faster disambiguation time. I believe the overall advantage for Lock&Twist is because of the perceived difficulty participants encountered in the initial ray lock of the AutoTwist technique, even with the -100 ms time that used for handling Heisemberg effects. Logically, the PointCursor is even faster than Lock&Twist but it cannot be used in AR HMDs because of the need of tracking

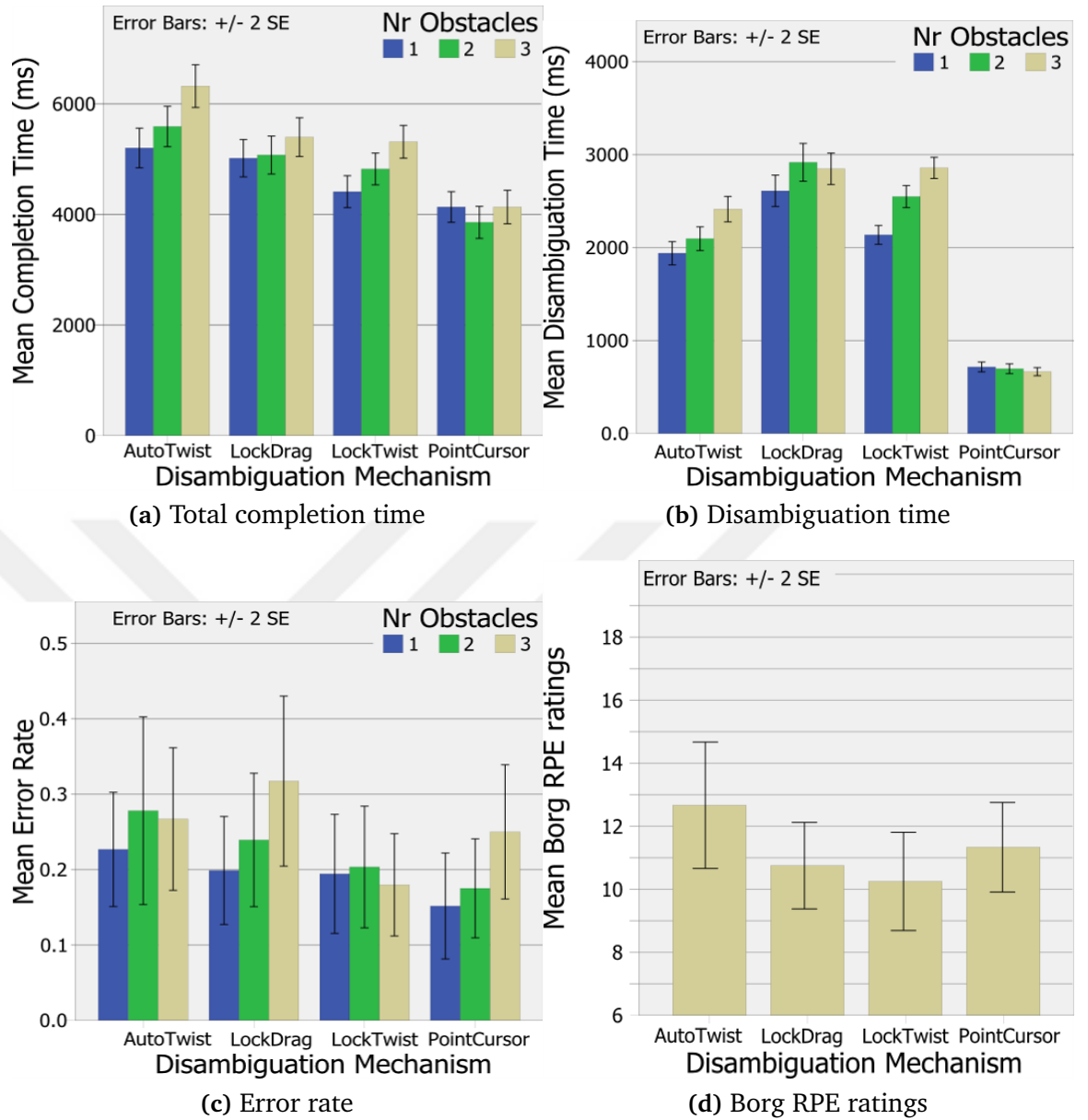


Fig. 3.11: Quantitative results the disambiguation techniques in selections with 1, 2 and 3 obstacles.

its spatial location, thus relying on external sensors. Nonetheless it serves as a good baseline (and hence why it was included in the study). With PointCursor, total selection time at around 4 sec, Lock&Twist adds 0.8 sec. This seems like a large amount of time, but considering that the user doesn't have to move in space it appears to be a reasonable trade-off. Therefore, using Lock&Twist is recommended when the number of targets to disambiguate is small. This is because having to navigate through a large list of objects by twisting the arm does not scale well, leading the users to twist their arms to uncomfortable

positions. A possibility to deal with scalability is a disambiguation mechanism combining Lock&Twist and Lock&Drag. Allowing the user to switch to the more time-consuming finger dragging navigation when the twist movement is inconvenient.

3.6 General Discussion

These studies highlight (a) the potential for using raycasting, even when full tracking is not available, as in outdoors environment; (b) the need to carefully select a raycasting origin; and (c) the ability to design suitable target disambiguation approaches even with an IMU sensor. From the results, I learn that:

1. IMU sensing alone, is a suitably good approach for target selection in AR HMD platforms. In the absence of external tracking, as is the case with mobile environments, raycasting is only slightly less efficient than a fully-tracked solution.
2. Raycasting using IMU data can originate at a location away from the controller (as when there is limited full-body tracking). An origin from the user's chin is a good candidate. This is particularly important for close-by targets, as was tested in this study.
3. Using small hand rotations, particularly wrist pronation and supination, are good design options for disambiguating targets laid out in 3D. In combination, a mechanism to switch into a disambiguation mode, such as the Lock&Twist approach, provides a suitable balance to mitigate unwanted movement and precise selection

3.6.1 Implementation Details

I implemented GyroWand for the EPSON Moverio BT-200 in Unity 4.6 for Android 4.0. Tracking data captured by the optical tracking systems (OptiTrack and Vicon) was received by a Unity application running on a desktop computer. The data was passed to the Moverio BT-200 via UDP packages using Unity's multiplayer gaming capabilities. To create the stereoscopic rendering in Unity I use the Stereoskopix FOV2GO Unity package created by the MxR group at the University of Southern California (USC). The experiments were carried out in two places. Experiment 1 was performed using an OptiTrack system

with cameras. Experiment 2 was carried using a Vicon tracking system with 8 cameras. To determine rotational movement I evaluated both Android's own orientation vector and Madgwick's x-AHRS [98]. Contrary to [8] the Android implementation appeared more stable and less prone to drift. Nonetheless, the EPSON Moverio BT-200 platform I used seemed to have frequent problems of sensor malfunction which required re-starting the controller for the IMU to provide clean data. I observed this problem on several EPSON Moverio BT-200 units I tested.

3.7 Conclusion

In this chapter, I explored the possibility of using accessible inertial measurement unit (IMU) sensing to bring a common interface for 3D mid-air interaction into mobile AR head-mounted display. Since its first proposal, raycasting, a line projecting away from the user's arm and into a scene, as an interface tool has witnessed numerous developments. Many of these rely on external tracking approaches, which are not readily available in mobile contexts for augmented reality applications on head-mounted displays. Instead, most commercially available devices are equipped with controllers that sense the relative 3D hand orientations using gyroscopes. I propose GyroWand, a raycasting interface for 3D object selection. This design address a number of challenges, such as identifying the source point of the ray and the ability to disambiguate the selection. GyroWand also uses a state machine to mitigate sensing and drift errors arising from such sensors, as well as involuntary 3D movements. Overall, the results suggest that raycasting can be adopted for HMDs with low-cost built-in tracking capabilities or simply with a smartphone for 3D selection tasks in terms of mobile context.

In future work, I will consider other disambiguating approaches, such as using a quadratic distribution function on the wrist tilt for more accurate disambiguation, and consider how such an approach affects interaction while on the move, an area less explored with raycasting solutions that have primarily been tested in indoor environments. I will also investigate how such an approach compares both in performance, as well as comfort, against natural user interfaces that involve natural arm movements. Finally, I will explore the potential of such an approach, when the controller is miniaturized for ease of use, such as when the IMU is available on a ring-based device.

4

Comparison of 3D Selection Techniques for a Singe-User Display

4.1 Introduction

See-through head-mounted displays (HMDs) offer Augmented Reality (AR) experiences by superimposing a virtual information layer on top of the real-world. Unlike earlier AR systems which relied on complex infrastructures such as external 3D positional tracking systems, new commercially-available AR devices are equipped with built-in control systems such as hand-held controllers, gesture systems, and simultaneous location and mapping (SLAM) which allow users to interact with the virtual content in any environments and without external and complex infrastructures. These new built-in tracking capabilities have lowered the cost and complexity of using HMDs, and have the potential to make these devices and applications a reality for everyday usages and activities.

However, despite of potential of the mobile AR devices their input technology is still limited, especially for the wide range of interactions required for spatially-located 3D objects. 3D interaction techniques developed for other environments such as 3D desktop displays and immersive 3D systems do not directly translate to HMDs because of limitations in display field of view (FOV), and the quality and tracking capacity of the embedded sensors. Moreover, the research about how the existing 3D selection techniques apply to the new AR devices is very limited. Despite some researchers having explored mid-air selection techniques [8] [182][48], and 3D techniques with the built-in sensors and hand-held

controllers of the devices (e.g., GyroWand [60], Myopoint [54]), no research has compared these selection techniques and their relative advantages for HMDs.

In the previous Chapter, I have investigated adoption of raycasting techniques for selection task in HMDs using built-in movement sensing. This chapter investigates how different existing 3D interaction techniques, virtual pointing and virtual hand techniques, apply to the unique circumstances of HMDs in mobile scenario. However, understanding the capabilities of current built-in display and sensing technologies on these devices, and determining how they influence selection performance and user preference must be examined.

Therefore, I explore a set of 3D selection techniques and compare them in mobile HMDs through a user study. The study compares five existing techniques that could work on HMDs in mobile setting with embedded motion and optical sensors: GyroWand [60], head cursor, gesture-enabled dwell, flick, and bimanual selection techniques. GyroWand is a ray-casting technique using the IMU of a handheld controller. Head cursor is another raycasting technique based on head rotation sensed by a built-in IMU within the HMD. Dwell, flick and bimanual selection are hand-based techniques with different confirmation mechanisms that enable users to control a 3D cursor. I studied how these five techniques perform in simple spatial object selection tasks. While the five techniques do not cover all aspects of spatial object selection, they involve different modalities, DOFs (degree-of-freedom), and limb movement ranges. I believe that exploratory a comparison offers valuable design insights for interaction techniques for arm's reach object selection in mobile AR settings for HMDs. In this comparative study, I explored how well users were able to acquire a set of arbitrary objects spatially located within arm's reach. I collected data about the task completion time, accuracy and arm fatigue. I also examined two types of reference frames, body and world, that could characterize the selection techniques and their performances [60]. The body reference enables content to be always at the same location in relation to the user's body, while the world reference enables content to remain fixed in space regardless user's location.

The work in this chapter presents the following contributions:

- An exploration of several 3D selection techniques for within-arms reach spatial object selection in mobile HMDs.
- A discussion of the best selection technique for spatial object selection in mobile AR. The results of a user study shows that head-directed 2D cursor

based on raycasting outperforms the other selection techniques in terms of speed and user preference in both reference frames; body and world.

- A discussion of the techniques and their major design characteristics specific for mobile HMDs.

4.2 Related Work

Latest optical see-through head-mounted displays (HMDs) require a set of optical and movement sensors aimed at providing spatial tracking of user's actions and input without relying on external infrastructures. In this section, I review these built-in tracking technologies in HMD, their use for spatial selection techniques as well as comparative studies on 3D selection techniques.

4.2.1 Sensing Technologies of HMDs

An important approach to track motions of HMDs relies on using cameras embedded on the device to determine the position of the viewer in relation to the real-world. These approach uses image processing techniques such as feature extraction and matching to build a model of the world. Implementation of these technique can be found on commercial SDKs such as Metaio [105] or Vuforia for Smart EyeWear [183]; simultaneous location and mapping (SLAM) using depth sensors [141] and traditional low-end RGB cameras. In some cases, the camera information can be enriched with rotational and inertial as provided by IMUs [206]. Besides using cameras for the purpose of positioning the HMD in the real world, the camera can also be utilized to track users' hands gestures and movements for mid-air interactions. The optical tracking capabilities (i.e. camera) of HMDs are nonetheless limited and, influence designs of interaction techniques which can be realized in HMDs. In the case of raycasting techniques, such limited tracking supports only raycasting from user's head [170] and from the head through the hand [130]. Another approach is using motion sensors which are not located on the head piece, instead located on a hand-held controller [36] in which an IMU in the controller allows user to control the ray with the device orientation [60]. For HMDs without a hand-held controller, the IMU data could be provided by external wireless hand-held controllers such as wands, rings or even accessible smartphones.

4.2.2 3D Selection Techniques

Selection with Virtual Raycasting

Raycasting has been widely studied for 3D user interfaces in virtual reality [8][21] [126]. In its default implementation, a virtual ray is casted from a point of origin, usually a tracked hand or hand-held controller, along a given pointing direction. When a trigger event is issued, the ray intersects several virtual objects – typically the object closest to the ray is selected. Raycasting is a technique for distant pointing which has limitations for object manipulations such as translation and rotation. Further, Raycasting often requires disambiguation mechanism among possible multiple targets particularly for densely populated virtual environments. Raycasting implementations vary in various aspects including how the ray is controlled. Controlling the ray requires a point of origin and a direction. These two values can be provided by tracking the position and orientation of a controller or wand [46] [126], the user hands [22] [104], the head [170], or a mix of hands and head [130] [73]. However, the most common approach relies on tracking a hand-held controller with a button for selection confirmation. Another important variation is the shape of the ray: it can be an actual line in 3D space [8], or it can have an aperture angle effectively configuring a cone [39]. Using a cone improves the selection of smaller targets and compensates for hand trembling and jitter, while increasing the need to disambiguate between possible targets.

Selection with Hand-controlled 3D Cursor

Previous studies showed that direct mid-air selection using the hands, in which visual and motor spaces are closely coupled, provides higher performance than indirect or remote selection techniques [32] [187] [90]. However, designing mechanisms for selection confirmation is critical since it significantly effects the performance of entire selection task. Therefore, numerous work has considerable attentions on this mechanism using typically physical button or touchpad, or voice command, dwell time, and quick hand gestures.

One approach uses a dwell time as the confirmation mechanism, where a click event is recognized if an input device (e.g., cursor) is kept in a steady state for a certain time threshold, even though it introduces a constant lag.

Using quick hand gestures is another approach to confirm a selection. Grossman et al. [46] proposed thumb trigger gesture where thumb finger moves in and out towards the index finger in mid-air object selection in a volumetric

Tab. 4.1: Summary of the comparison of selection techniques

Compared Techniques	Cursor type	Target range	Display		Evaluation
			Type	Dim.	
Head, Hand-directed, Head-hand directed ray cursor, and Image-plane [86]	Ray, 2D cursor	Remote	Large Display	2D	<i>Image-plane: The fastest, the most accurate, no fatigue Head-hand directed ray cursor: The slowest, the least accurate, a lot of fatigue</i>
Eye-directed image-plane, virtual hand [170]	Ray, 3D cursor	Arm-reach, Remote	VR HMD	3D	<i>Eye-directed image-plane (remote): The fastest, no fatigue Eye-directed image plane: No fatigue</i>
Image-plane, virtual hand [190]	3D cursor, 2D cursor	Remote	Stereoscopic Display	3D	<i>Image-plane: The fastest</i>
Hand-directed ray, virtual hand [46]	Ray, 3D cursor	Arm-reach	Volumetric Display	3D	<i>Hand-directed ray: The fastest</i>

display. Vogel et al. [182] compared this with their AirTap technique, which is similar to our index finger movement when clicking a mouse for large-size 2D display interaction. Thumb trigger gesture has advantage of kinesthetic feedback since it still touches the side of hand. As shown by Wang and MacKenzie [188] that absence of these physical surfaces significantly degrades performance of object selection. Nevertheless, Vogel et al. [182]’s results showed that there is no significant difference in trial performance time or error rate between these two different click gestures.

4.2.3 A Brief Survey of Selection Techniques

Table 4.1 summarizes main elements of existing studies of comparisons on 3D selection techniques. These studies, alongside with the one presented in this paper, can help readers understand the tradeoffs of existing selection techniques. Unlike others, this study focuses on the use of the selection techniques in the particular case of mobile HMDs. Table 4.1 shows my classification based on main characteristics (i.e. display type and input type), performance, and usability of selection techniques. As general goals of interaction techniques, in this survey, I focused on speed, accuracy and fatigue as the parameters of effectiveness and efficiency.

Lee et. al. [86] compared four selection techniques such as head-directed, hand directed and head and hand-directed raycasting, and image plane techniques for a large 2D display setting. Their results showed that image-plane technique outperformed the other in terms of speed, accuracy and hand fatigue.

Besides, the head-directed raycasting performed the worst performance in terms of speed, accuracy, and hand fatigue.

Tanriverdi et al. [170] compared eye movement-based 2D cursor, which uses dwell-time selection confirmation for a focused object, to conventional 3D cursor for arm's reach and remote selection tasks. Results showed that eye movement selection was faster than hand-based pointing for only the remote objects, which required to user to move forward to select them. Also, users did not spend extra physical effort to select the objects with eye-movement-based interaction. Ware et al. [190] compared one-eyed image-plane using 2D cursor with the virtual hand using 3D cursor for VR HMD. Image-plane technique was faster than virtual hand. Grossman et al. [46] confirmed that raycasting is faster than virtual hand for volumetric displays where the objects are within arm's reach.

While a set of comparative studies have been done as shown above; they are generally focused on large 2D display, stereoscopic displays, VR HMDs, or volumetric displays. None of the studies focused on arm's reach selection in HMD settings. For recent HMDs, it is therefore important to revisit the primary and fundamental questions to find what technique is the most suitable to select spatial object in arm's reach in terms of speed, accuracy, and usability in mobile HMDs scenario.

4.3 Selection Techniques for Mobile HMDs

I implemented and compared five techniques having that can be realized using sensors (image and movement sensors) embedded in a mobile HMD. Our goal is to understand how efficient they are for object selection within arm's reach in mobile AR settings. To find out suitable interaction techniques, I pick up and explore five potential interaction techniques.

The techniques are selected from two different approaches. GyroWand and head cursor are ray-based, and use movement sensors (IMU) on the controller and head piece of the HMD respectively (Figure 4.1 at left and right). Other three techniques (Figure 4.2) are based on 3D cursor controlled by the user's hand. This study focuses on the performance and user's preference of the two ready-to-use raycasting, and three hand-controlled 3D cursor techniques. In order to obtain more reliable data than the one form embedded sensors in the HMD devices available today, I used external trackers to emulate these techniques to validate their performances rather than exploring their real-world

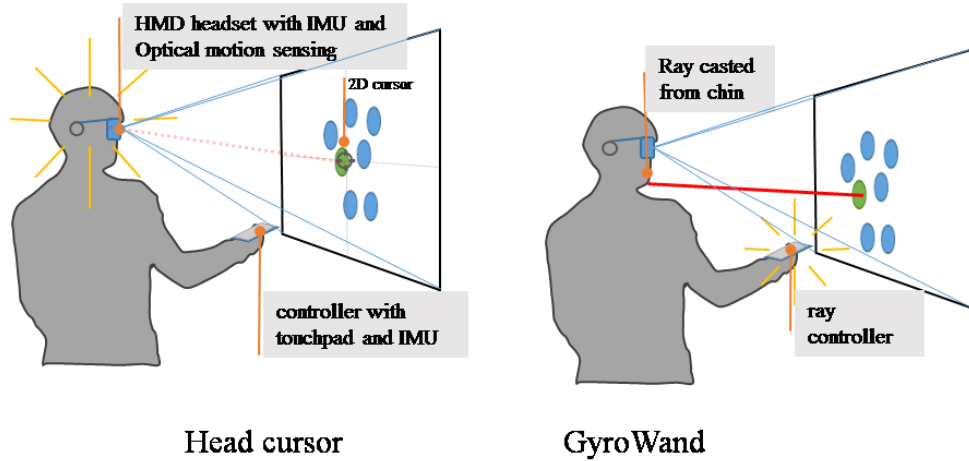


Fig. 4.1: (left) Head cursor: user looks straightforward and stationary cursor appears at the center screen. User rotates his/her head to hover a target with the cursor. The ray direction is controlled using the IMU on the head piece. (right) GyroWand: The ray origin is the user's chin (See section 3.4.1)

issues.

GyroWand and head cursor use only rotational data to acquire spatial object: GyroWand uses vertical and horizontal rotations of the hand-held controller; head cursor uses head rotations. All hand-controlled techniques require data about the location of the hand in space as input, enabling users to acquire spatially located objects through a 3D cursor with different selection confirmation mechanisms.

4.3.1 Raycasting Techniques

Raycasting techniques have been previously studied and found to be efficient for spatial object selection. For mobile HMDs raycasting techniques can be realized by directing a ray originated at a predefined location in virtual space, and use rotational data acquired from an IMU in the user's hand-held device. However, this naive rotational data typically cannot be directly mapped to ray orientation because the rotational data gathered from IMUs has several well-known problems: noise, sensor drift, and axis mapping. Therefore, raycasting techniques interpret IMU rotational data using a state machine and compensate for drift/interference [60].

Head cursor

Head cursor is a raycasting technique that enables to interact with the 2D projection of remote objects on the user's viewing plane as shown in Figure 4.1

at right. The object crossed by the cursor is selected by casting an invisible ray from the eye-point. This technique particularly benefits from the recent available IMU sensor on the head piece of HMD. As the user rotates his/her head the projected cursor moves on 2D plane. It requires a selection confirmation to make a click on a target for which I used a touchpad on the hand-held controller. When a selection is confirmed by tapping on the controller, the crossed virtual object with the closest 2D projection to the cursor is selected.

GyroWand

I used GyroWand [60], a state-of-the-art raycasting technique designed for mobile HMDs. GyroWand introduces fundamental design differences from traditional raycasting. GyroWand does not initiate raycasting from the user's hand, but rather from any spatial coordinate (e.g. chin, shoulder). In this study, GyroWand directs a ray originated at user's chin as on-body location in virtual space using hand-held controller's rotational data acquired from built-in IMU as shown in Figure 4.1 at right.

4.3.2 Hand-controlled 3D Point Cursor Techniques

Hand-controlled techniques are also explored using 3D point cursor (i.e. point cursor techniques), since selecting objects with the fingers seems natural and direct way to interact with objects within arm's reach. As I described earlier, my set-up in the experiment used a set of external 3D motion trackers instead of self-contained sensing capabilities as a way to achieve stability and reliability. Nonetheless, I can expect self-contained gestural tracking to be part of future HMDs as showcased in the upcoming devices such as Atheer One and the META 2.0 devices. I use a 3D motion tracking environment using an external optical tracking system with infrared cameras and reflective markers on the fingers to avoid technical and methodological constraints caused by implementation.

Moreover, because the way of selection confirmation is critical in this approach, I investigated two types quick finger gestures (such as dwell time and flick) to avoid problem of carrying a hand-held controller, and using both hands as manual confirmation with the controller (Flick in Figure 4.2 at middle). I introduce three techniques for selection confirmation below.

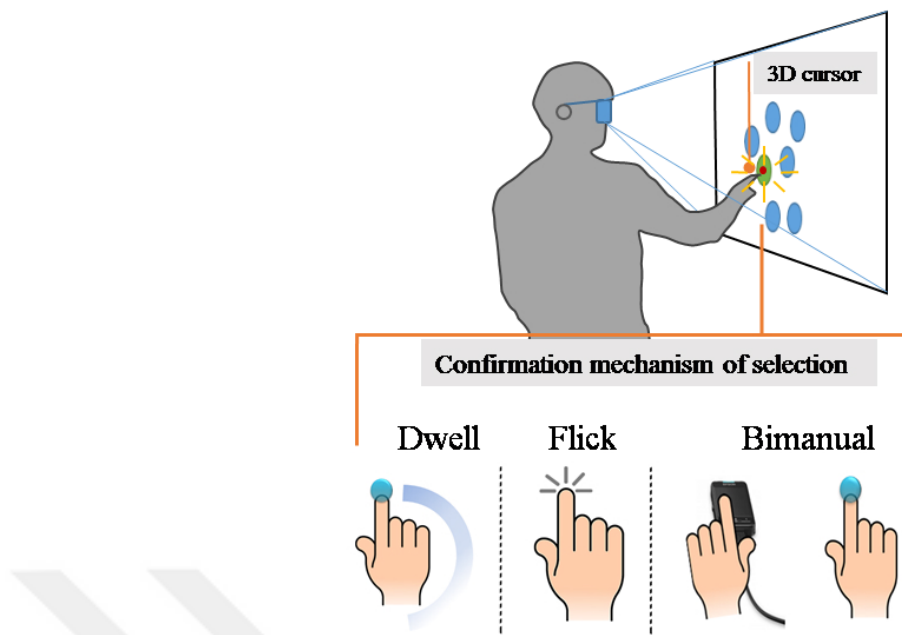


Fig. 4.2: 3D cursor: hand movement-based techniques using optical sensing with different selection confirmation mechanisms: Dwell, Flick, and Bimanual.

Dwell

This technique is one of the most commonly used gestures, which uses dwell time interval as confirmation as shown in Figure 4.2 at left. Users' hand/finger remains nearly motionless within a certain predetermined small range of space for a short time. This time interval must be properly considered in advance because it might induce unintended commands if the time interval is too short or too long, an issue known as the Midas touch problem [199]. Moreover, it can cause delays of interaction and causes discomfort if the interval is too long. Based on previous studies [115] [157], an optimal dwell time of 750ms is recommended for novice users; therefore, I use this dwell time in this work.

Flick

This technique (Figure 4.2 at center) consists of down and up movements of the finger. Since the hand is also pointing, I aimed to design this confirmation mechanism in a way that minimizes hand movement side effects. Because of the interconnectedness of tendons in the hand, movement in a finger affects the others. Therefore, I attached a marker to the index fingertip to detect flick gesture, and a 3-point marker on the hand to track finger position by offsetting its virtual position to index fingertip. Velocity (v) and acceleration of the user's finger provide cues to detect the click gestures; therefore I have built

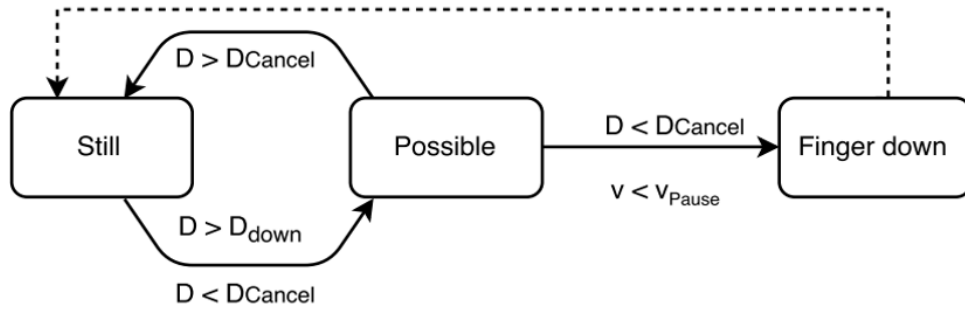


Fig. 4.3: Recognition state machine for flick gesture.

a recognition state machine similar to [182] using these parameters as shown in Figure 4.3, where finger velocity is marked by v , and D is distance moved in 200ms; D_{down} and D_{cancel} are thresholds to begin a click down and cancel actions, respectively. D and v are measured along axis of finger movement.

Bimanual: Tapping with Non-dominant Hand

Another possible approach for confirmation is to use dominant hand to move the cursor and the non-dominant hand for confirmation with hand-held device (Figure 4.2 at right). A user points at an object with his/her index finger of dominant hand, and confirms the selection by pressing a button or tapping a controller with his/her finger of non-dominant hand. Although this approach requires carrying hand-held controller, it is expected to reduce the interaction time because the dwell time is not necessary.

4.4 User Study: 3D Object Selection Task

I conduct a controlled user study to explore five techniques that rely solely on tracking technologies built into commercially-available HMDs in 3D selection tasks. I conducted a controlled user study to explore five techniques that rely solely on tracking technologies built into commercially-available OHMDs in 3D selection tasks. The goal of this study is to make an initial assessment of the possible interaction alternatives and that further research is needed to identify the particular aspects in which they differ and the strength of their impact - like the nature of the required movements.

The experiment aimed at answering the following fundamental questions:

- How fast and accurately can users select a set of arbitrary objects spatially located within arm's reach with interaction techniques?
- How do reference frames affect the task performance?
- How do techniques affect task workload index and user preferences?

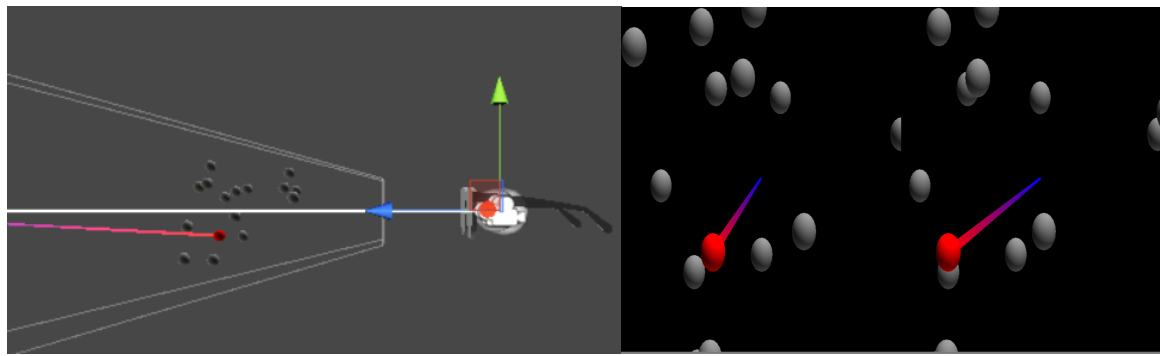
I focused on the two reference frames rather than using a fixed content in the user's limited FOV because these reference frames enable users to observe beyond their narrow FOVs. Reference frames also provide users the ability to explore a larger 3D virtual environment that could contain multiple larger objects.



Fig. 4.4: A participant using an HMD with markers for optical 3D motion tracking, having a controller for selection in his hand. Here, he is performing the task with a ray-based technique using IMU.

Apparatus - For the mobile AR setting, I used the same apparatus in previous chapter (See 3.4.1).

Subjects - 12 participants (4 female) volunteered, ages 21-32 (mean=24.9, stdev=3.56), all right handed. Five participants had previous experience with



(a) Objects of 1 cm width are randomly located 40-65 cm in front of the HMD (b) Stereoscopic rendering of the user view

Fig. 4.5: Rendering of user's view

head-mounted displays, four had experience with virtual reality, three had experience with mid-air interaction and none of them had experience with raycasting techniques.

Task - Figure 6.3.a and 6.3.b respectively show objects located in front of the HMD, and a pair of a stereoscopic view volume (cube-shaped) for target selection tasks. For each session, participants had to select randomly located 16 sphere-shaped targets in the cubic view volume. All targets were initially shown in grey and became highlighted in light green when they encountered a ray or a cursor. The target to be selected next is shown in red. When selection target is crossed by the cursor/ray its color changes to brighter red. On selection the system shows the target in magenta, and highlights another object in red as the new target. As Figure 6.3 shows, a line colored red-to-blue indicated the location of the next target, guiding the participant when it was out of the FOV. The session ended when the participant completed 16 target selections. An error is indicated when a wrong object or no object is selected.

Design - Independent variables were interaction technique and reference frame. I used a 5x2 within-subject design to investigate user performance. I considered five techniques (two raycasting: Head cursor and GyroWand; and three hand-controlled 3D point cursor: Dwell, Bimanual, and Flick). I randomly located 3D objects in a virtual cube 50cm away from the HMD but on different reference frames. While my primary interest is to explore 3D selection techniques for mobile HMD, I were also interested in the effects of reference frame of the technique. Because, for 3D objects within arm's reach the reference frame makes a difference as objects reacts differently to user movement. I evaluated two reference frames: body and world [59]. On the body reference

frame, content was always at the same location in relation to the body. For example, content positioned on the right of the body will always be visible when the user turns his head to the right, but not when he looks forward. On the world reference frame, content remains fixed in space irrespective of the user location or gaze direction; I located the objects at the center of the tracking cube and 1.5 meters high, participants stood 50cm away. Participants were asked to hold the controller in their dominant hands (Figure 4.4). However, when the confirmation was required in case of using a finger based technique, they held the controller in their non-dominant hands.

The experimenter demonstrated the task and then asked the participants to train for 10 minutes with each technique. The training session consisted of at least 20 selections for each technique. The orders of selection techniques and reference frames were counter-balanced using a Latin-square design. With a total of $5 \times 2 = 10$ conditions per participant and $16 \times 2 = 32$ selections per condition, I registered $32 \times 10 = 320$ selections per participant, 3840 selections in total. All participants completed the experiment in one session lasting approximately 50 minutes.

Measures - For each trial I recorded task completion time, selection errors, and cursor arrival times to targets. Completion time is the time elapsed from cursor departure until the object is selected. Selection error is the count that indicates the number of wrong object selections and no object selection. First arrival time is the time elapsed from cursor departure until it first arrives to the target object (before triggering selection), last arrival time is the time elapsed from cursor departure until the last arrival to the target (before triggering selection). A user might arrive at the target object (first arrival), overshoot it, arrive again (last arrival) and then trigger selection. For each condition I collected Nasa-TLX workload index and Borg RPE (rating of perceived exertion).

4.4.1 Results

I removed outliers (5% of the data set) after 3 standard deviations of completion time. I carried out ANOVA and post-hoc tests with Bonferroni corrections (only for interaction technique) to analyze the measurements.

Completion time - As shown in Figure 4.6a, I found a main effect for interaction technique ($F_{4,44} = 15.161, p < 0.001, \mu^2_{partial} = 0.579$). I did not find main effect for the reference frames ($p = 0.829$) while there was no significant interaction of controller x reference frame.

Post-hoc tests showed significant differences between all techniques (p

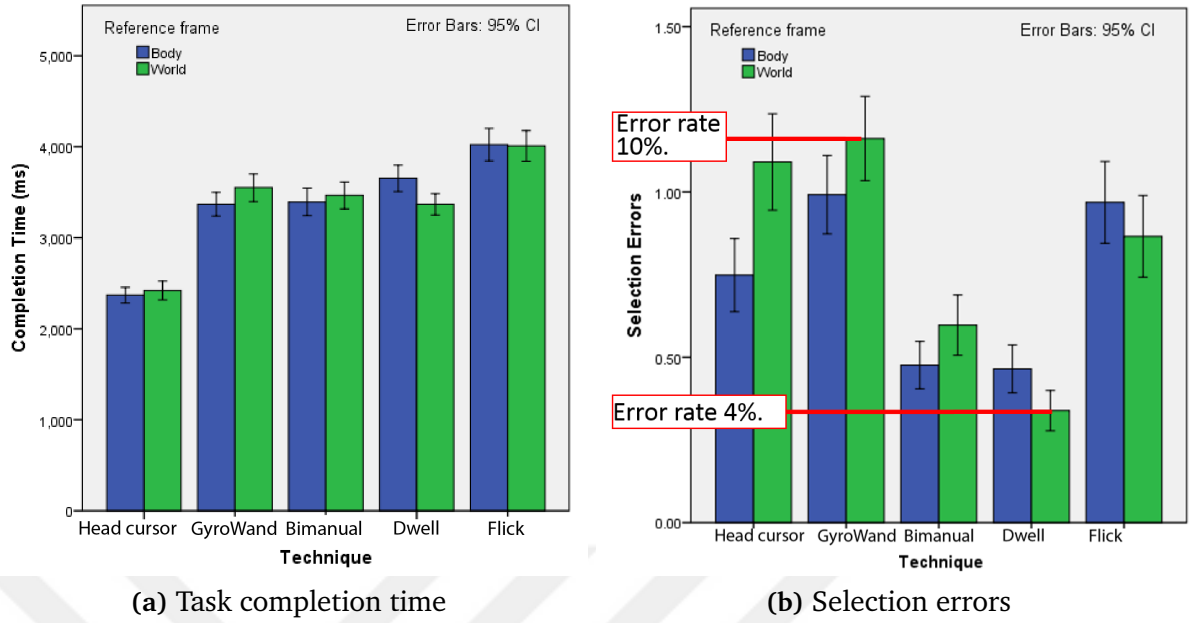


Fig. 4.6: Quantitative results

< 0.001) except, between Bimanual and GyroWand ($p = 1.0$), and between Bimanual and Dwell ($p = 1.0$). Head cursor had the fastest completion time at 2.395 sec (stdev = 0.939), followed by bimanual, GyroWand and Dwell at 3.429 (stdev = 1.445), 3.458 (stdev = 1.373), and 3.510 (stdev = 1.295) sec respectively. Flick was the slowest completion time at 4.015 sec (stdev = 1.572).

Selection errors - As shown in Figure 4.6b, I found a main effect for technique ($F_{4,44} = 11.333, p < 0.001, \eta^2_{partial} = 0.398$) and an interaction effect between technique x reference frame ($F_{4,44} = 3.369, p = 0.017, \eta^2_{partial} = 0.232$). Results did not show main effect for reference frame ($p = 0.054$). A change in the selection technique explains 57.9% of the change of selection speed (total time $\eta^2_{partial} = 0.579$) and 39.8% of the change of selection errors (selection error $\eta^2_{partial} = 0.398$). Moreover, a change in the reference frame explains 0.4% of the change of selection speed (total time $\eta^2_{partial} = 0.004$) and 29.4% of the change of selection errors (selection error $\eta^2_{partial} = 0.294$).

Borg RPE - Figure 4.8 shows the results of Borg RPE. I found a main effect for technique ($F_{4,44} = 25.937, p < 0.001, \eta^2_{partial} = 0.702$) and the post-hoc tests showed significant differences between all techniques ($p < 0.001$) except, between Bimanual and GyroWand ($p = 1.0$), and between Bimanual and Dwell ($p = 1.0$). Head cursor presented smallest physical exertion at 8.50 (stdev = 3.35), and followed by GyroWand at 10.792 (2.55). Results did not show a main effect for reference frame ($p = 0.927$).

First arrival-last arrival- Completion time alone, does not provide information regarding ray/cursor movements during the interaction. According to Woodworth's two-component model [196], if a target is not acquired with cursor due to either undershoot or overshoot on the first try, then the user performs iterative slow and more accurate movements until acquire it, known as corrective phase. The difference between the first and the last arrival times approximates corrective phase in our study. Therefore, we examined this differences to identify behavior of the cursor movements during a selection task for each technique. Figure 4.7 shows this difference relative to the technique. Results showed a main effect for technique ($F_{4,44} = 9.22, p < 0.001, \eta^2_{partial} = 0.456$.) Post-hoc tests showed significant differences between all techniques ($p < 0.001$) except, between Bimanual and Dwell ($p=0.42$). Bimanual and Dwell were the techniques with the least difference at 0.392 (sdev =0.027) and 0.317 (sdev = 0.026), followed by Head cursor at 0.490 (sdev =0.026), GyroWand at 0.762 (sdev =0.027) and Flick at 0.957 (sdev =0.029).

Here, first and last arrivals are time of arrivals of the cursor at a target before selection. I analyzed the differences between first and last arrival times for all the techniques. Results showed a main effect for technique ($F_{4,44} = 9.22, p < 0.001, \eta^2_{partial} = 0.456$.) Post-hoc tests showed significant differences between all techniques ($p < 0.001$) except, between Bimanual and Dwell ($p=0.42$). Bimanual and Dwell were the techniques with the least difference at 0.392 (sdev =0.027) and 0.317 (sdev = 0.026), followed by Head cursor at 0.490 (sdev =0.026), GyroWand at 0.762 (sdev =0.027) and Flick at 0.957 (sdev =0.029).

Subjective Results

Participants' subjective workload assessment and ranking, shown in Table 4.2 and Table 4.3, explained the quantitative results, clearly showing that participants found the 3D point cursor techniques mentally demanding, tiresome, slow, and effortful, (See Table 4.2, shown with asterisk). NASA-TLX shows that mental load, physical load, effort and frustration with Head cursor and GyroWand are much smaller than the finger techniques. Assessment of the head cursor and GyroWand interface was positive with general preference for Head cursor (See Table 4.3). The head cursor was ranked top by eight participants (*Mean* 1.5), while GyroWand was favored by three (*Mean* 2.58). The main effects of technique on ranking are shown in Table 4.4. Moreover, some participants stated that they liked the way of controlling the ray with low and small

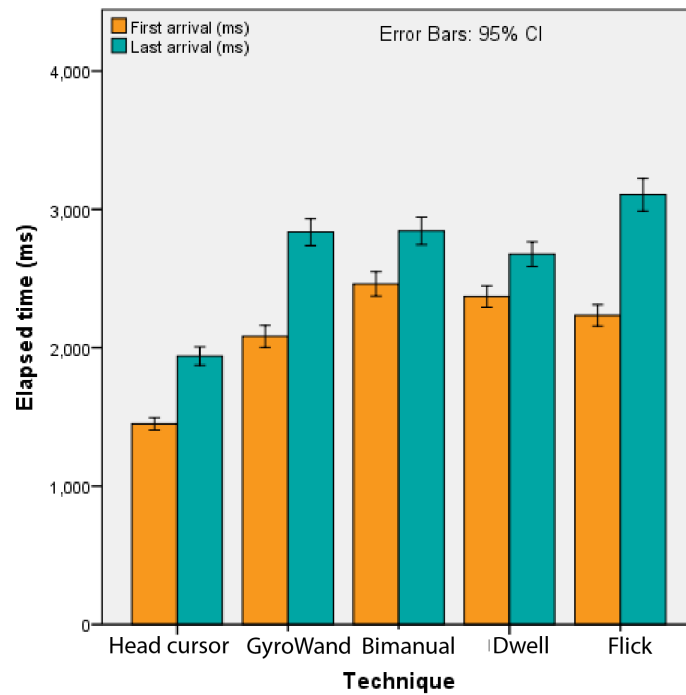


Fig. 4.7: First-last arrival time to target

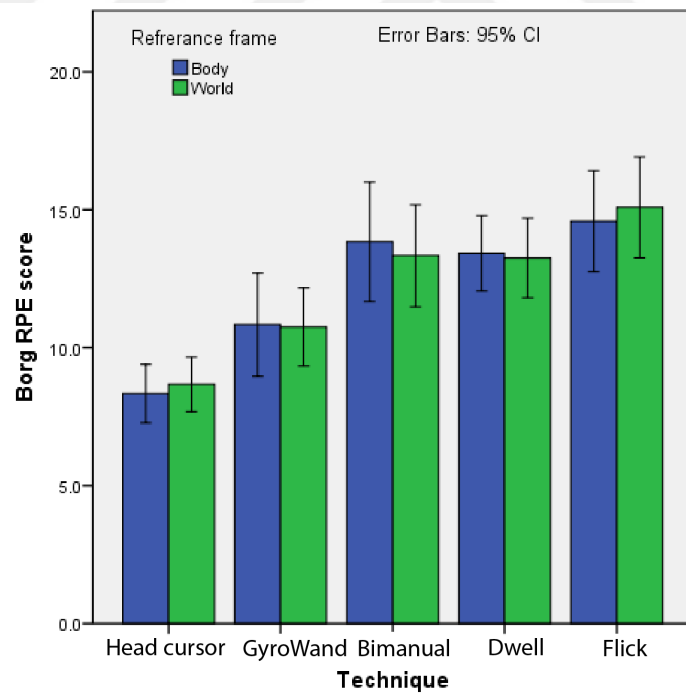


Fig. 4.8: Borg RPE score

hand movements ("GyroWand is like controlling fancy Lightsaber in Star Wars"), but found it harder to control than Head cursor. The 3D point cursor techniques are physically demanding (Table 4.2, shown with asterisk); however, provides

ease of control (Table 4.3).

Tab. 4.2: Mean (and standard deviation) subjective preference assessments for each technique

Nasa-TLX (1-21)	Head cursor	Gyrowand	Dwell	Bimanual	Flick
Mental load (Low - High)	4.75 (3.34)	6.79 (4.32)	7.45* (4.2)	7.33* (5.4)	9.04* (5.0)
Physical load (Low - High)	4.79 (4.1)	7.08 (4.9)	12.04* (4.6)	13.3* (5.09)	14.29* (4.3)
Temporal load (Low - High)	4.29 (2.5)	6.08 (3.41)	7* (3.5)	8.62* (4.37)	10.20* (4.7)
Performance (Perfect - Failure)	4.04 (2.62)	6.12 (4.53)	6.04* (3.9)	7.75* (4.95)	9.41* (5.0)
Effort (Low - High)	4.83 (4.04)	7.04 (4.5)	9.79* (4.7)	10.87* (5.10)	12.16* (4.8)
Frustration (Low - High)	4.87 (3.8)	6.70 (4.8)	7.04* (5.3)	7.83* (5.49)	9.125* (5.9)

4.4.2 Discussion

Experimental Results

Head cursor outperformed the others in terms of selection speed (Mean 2 sec.). 3D point cursor techniques, except Flick, were more accurate than raycasting techniques. In terms of tradeoff between speed and accuracy, Flick and GyroWand may provide the least benefit since they are both slower and less accurate.

With GyroWand, virtual objects are selected by pointing directly to themselves with a casted ray; however, with head cursor technique they are selected by pointing to their screen projection through a 2D cursor. Therefore, while objects in different depths have similar screen projections, GyroWand required more limb orientation than the head controller to select them (See Figure 4.9).

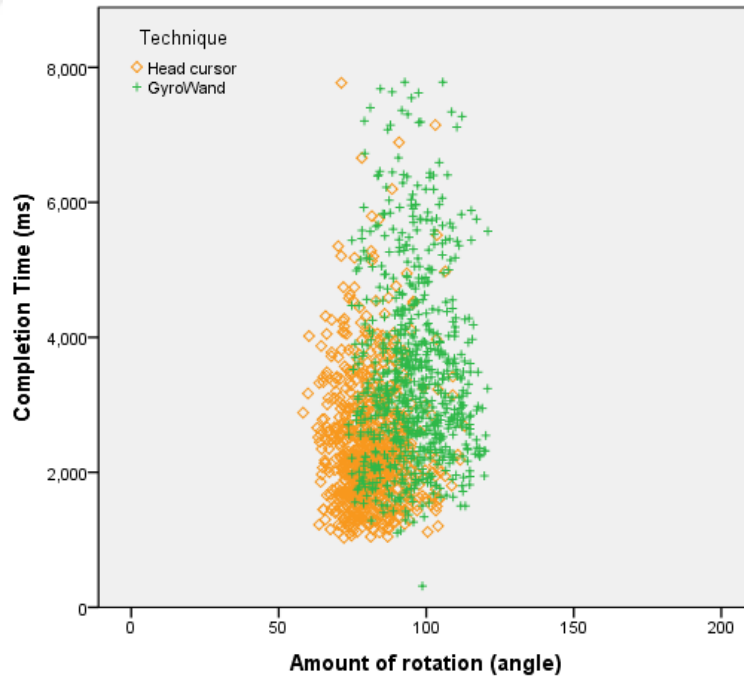
Tab. 4.3: Mean (and standard deviation) of ranking for each technique

Questionnaire(1-5)	Head cursor	Gyrowand	Dwell	Bimanual	Flick
Overall preference (High - Low)	1.5 (0.6)	2.58 (1.44)	3.25 (0.9)	3.5 (1.11)	4.16 (1.14)
Easy-to-understand (Disagree - Agree)	4.25 (1.01)	4 (0.91)	4 (0.91)	4.5 (0.64)	2.75 (1.65)
Well assisted (Disagree - Agree)	4.16 (1.28)	3.75 (0.92)	3.25 (1.01)	3.5 (0.76)	2.66 (1.31)
Easy-to-control (Disagree - Agree)	4.33 (1.1)	3.22 (1.17)	3.75 (0.92)	3.66 (1.02)	2.5 (1.38)

Tab. 4.4: Main effects of techniques on ranking

Main effects of technique on	Significance difference between
Overall preference ($F_{4,55} = 9.544, p < 0.001, \eta^2_{partial} = 0.410$)	Bimanual and Head cursor ($p = 0.001$), Dwell and Head cursor ($p = 0.004$), GyroWand and Flick ($p = 0.012$), Flick and Head cursor ($p < 0.001$)
Understandability ($F_{4,55} = 4.461, p = 0.03, \eta^2_{partial} = 0.245$)	Bimanual and Flick ($p = 0.03$), Head cursor and Flick ($p = 0.016$)
How assisted ($F_{4,55} = 2.973, p = 0.027, \eta^2_{partial} = 0.178$)	Flick and Head cursor ($p = 0.019$)
Controllability ($F_{4,55} = 3.868, p < 0.05, \eta^2_{partial} = 0.220$)	Flick and Head cursor ($p = 0.04$)

Another reason, possibly, when using head cursor in given reference frames, a user rotates his head towards the targets to select them regardless the target remains whether in the FOV or not. However, using GyroWand when the target remains out of the FOV requires user to do a two-step operation; user needs to first rotate his head to spot the target, and then direct the ray by using the controller.

**Fig. 4.9:** Amount of angular displacements of Head cursor and GyroWand techniques from one selection to the next

Traditional raycasting has previously been shown to outperform virtual hand using 3D cursor [19] for 3D environments such as volumetric display [46]

where 3D content was also within arm's reach. However, in this study GyroWand (Raycasting), which is faster than a traditional hand-directed raycasting [60], and hand techniques (3D cursor) performed similar speed. While motor and visual space are superimposed in this settings, therefore users can only rely on proprioceptive feedback, study in [46] decoupled these spaces due to using a volumetric display. This decoupling makes the proprioceptive feedback insufficient for users during the task and requires also visual feedback [8], which might require additional time, resulting slower performance with 3D cursor.

In this study, the difference between the first and the last arrival times approximates corrective phase of a goal-directed movement. This phase of the movement corrects initial rough movement to select the target with slower velocity. While both GyroWand and head cursor are ray-based, required times for correction phase are significantly different. I think one reason for this is that input error (e.g. jitter from head/hand, or IMU) becomes larger when using GyroWand, because it presents a larger orientation amount for selections (See Figure 4.9). To reduce this (if IMU noise-induced), while the user is refining the selection with small movements, and therefore generating less errors, the ray apex can be narrower automatically in the corrective phase. However, further study is required to find whether the jitter differences between group of muscles in head (neck) and hand cause this difference or not. However, for the Flick, I think the participants had difficulties with controlling its mechanism as they mentioned in the questionnaire. Regardless having experience with HMD, hand-controlled mid-air interaction, and virtual reality, eleven participants (91.6%) chose raycasting techniques for 3D selection techniques. This could be because while raycasting techniques require low and small movements of limb. Participants commonly suffered from arm fatigue in the all hand-controlled 3D cursor techniques (See Physical Load in Table 4.2).

In the all hand-controlled 3D cursor techniques, participants commonly suffered from arm fatigue (See Physical Load in Table 4.2). Because, the repeated use of a mid-air hand is likely to cause fatigue. As I previously stated that hand-controlled 3D cursor is selected because of its naturalness and intuitiveness, which can be a powerful interaction way. However, the results show (despite using high fidelity tracking) that it is not true. Even if I will have an ideal vision-based hand tracker around HMDs in the near future, such a hand-controlled 3D cursor would not be a well-accepted input method due to its fatigue, and social issues in outdoor environments. These problems could be generalized and common discussion point in direct mid-air interaction

for mobile setting. Therefore, I can infer that limiting the amount of limb's physical movements is an important consideration when designing interaction techniques. For this reason, IMU-based interaction technique is an alternative that reduces such movements required for interaction as it simply requires minor tilting of the head or hand.

This study highlights the comparison of selection techniques in terms of speed, accuracy, subjective workload and users' preference as for HMDs in outdoor environments. From these results, I learn that:

1. Using IMU sensing only, is a suitably good approach for object selection in mobile HMD platforms. In the absence of external tracking, as is the case with mobile environments, head-directed 2D cursor is an efficient pointing solution.
2. Comparing with hand wrist orientations, using head rotations is a good design option to select targets laid out in 3D in HMDs, and is preferable when some objects located out of user's FOV.
3. Overall, raycasting-based interaction techniques that reduce the amount of limb movements required, are still good to interact the objects within arm's reach with a mobile HMD.

4.4.3 Design Consideration

Participants had difficulty selecting spatial targets with hand-based techniques (3DOFs) within arm's reach, and performed slower selection time than ray-based techniques (2DOFs). However, the translational motions of hand resulted higher accuracy than the rotational motions of the ray-based techniques. Hand movement-based techniques are generally more accurate than particularly, 2D plane (head cursor) interface in contrast to findings in [29]. Although I explore 3D object selection within user arm's reach, both head cursor and GyroWand techniques provide selection of targets beyond arm's reach in the reference frames. Since the small movements of these two techniques are less physical demanding and fast, therefore less accurate for small remote targets, they could be probably preferable when interacting with larger targets in relatively large AR environments. Instead of direct hand-controlled 3D point cursor, constrained hand movements, such as arms-down mid-air techniques with higher control-display (CD) gain, using sensors mounted on the thigh, may deliver low arm fatigue and social acceptance issues for hand-controlled 3D

cursor techniques [54]. However, value of CD gain must be sensibly considered because it might increase overshooting, negatively affect accuracy because of the reduced distances in motor-space [93].

4.5 Conclusion

This chapter presents the results of a formal experiment comparing five mid-air selection techniques to interact with virtual spatial objects within arm's reach in personal optical see-through head mounted displays (HMDs). Selected techniques rely solely on tracking technologies built into conventional commercially-available HMDs, enabling interaction in mobile scenario where the devices do not need external tracking environments. I investigated the effectiveness and efficiency of five techniques for arm's reach selection: two are based on raycasting using built-in IMU sensing, and three are based on a hand-controlled 3D point cursor using gestural tracking on a HMD. Results showed that raycasting techniques using head rotation facilitates selection for speed, and ease of use in mobile HMD settings. In addition, hand-controlled 3D point cursor techniques are more accurate; however, presented high levels of arm fatigue in selecting objects. I have learned that reducing amount of limb movements, even for objects within arm's reach, is an important consideration when designing mid-air selection techniques for HMDs. I believe that these findings could be useful for designing future interaction techniques in personal self-contained HMDs.

In future work, I will consider to study whether the jitter differences between group of muscles in head (neck) and hand cause the difference between the head cursor and GyroWand techniques.

5

3D Mid-air Selection and Manipulation Techniques for 3D Multi-User Collaborative Display

5.1 Introduction

There is a growing body of research into 360-degree viewable 3D displays. Such systems provide viewpoint-corrected, stereoscopic 3D graphics to simultaneous users around the display, without the need for eyewear or other user instrumentation. These technologies also tend to provide depth cues such as divergence, correct eye-accommodation and horizontal motion parallax – the latter without the need for head tracking – which provide a perception of depth even for those without stereopsis. They have been shown to improve viewer perception of 3D when compared with stereo-only displays. Interactive displays are becoming standard to share displayed content by multi-users in fields such as industrial designing, game and movie industries, medical imaging, and so on. Such displays are often designed as tabletop forms for creating face-to-face workspaces and providing high accessibility of the displayed content. Multi-user interactive 3D displays, which allow each of the users to have an individual view are also becoming popular; however, most of them have significant limitation concerning direct touching to the 3D content because of its glass canvas or other optical devices to generate 3D image (e.g., [48] [203]). Such displays must be designed in such a way that gives users the opportunity to share understanding of the 3D content with direct indication [167]. To realize this opportunity, a tabletop 3D display, IllusionHole [79], has been explored which allows multiple

users to share a 3D content and directly indicate specific points on it with their fingers. It provides a distortion-free 3D content at exactly the same position from each of the viewpoint of multiple users; therefore, it can be used in many cooperative work.

Nevertheless, the display configurations of the system have bulky components under the table, such as either a huge 3D projector or a set of projectors for stereo-view with a mirror for bottom-projection. These cause the system to be a big enclosure where users have to be standing postures, and hinder flexible design challenges. In daily life, we often sit while working cooperatively; therefore, a more compact enclosure is required to achieve flexible designs for variety of multiple-user face-to-face inter-action. Moreover, we use our hands and fingers to directly interact with physical objects within arm's reach in daily life, therefore, hand-based direct interaction is essential to grab and manipulate the virtual objects. Because of working in a real 3D workspace, one's manipulation can be precisely observed by others from their own viewpoint.

In order to address these requirements, I designed a novel thin and compact IllusionHole configuration using a 3D TV for a typical collaborative workspace where users could sit around the table. Hence I reconfigured the design parameters for the proposed system to provide it with better collaborative observation. I also implemented direct object manipulation for the multi-user display system. By doing this, while a user is manipulating the virtual objects, others can share it from their own point of view.

Advances in tracking and display technologies create new opportunities for remote interactive collaboration. One particular instance of those new opportunities is the sharing of physical objects, allowing both local and remote partners to modify their properties. For example, a skilled designer sculptures a 3D model for a new device, while a group of potential clients at a remote location interactively observe and interact with its rendered 3D virtual shape at the same time. The changes performed by the designer are shown to the clients in real time, and the modifications made by the clients are transferred to the physical object. With such cooperative capabilities, remote collaborators can directly influence the details of new designs during the design process as if they were collocated. Therefore, I have utilized proposed interactive 3D tabletop display to create remote collaborative interaction: Coupled-clay, a bi-directional 3D collaborative interactive environment which enables a user to design and manipulate a 3D physical clay object whose shape and position are precisely tracked and remote users to observe and interact with its rendered

3D virtual shape in real-time. Coupled-clay is made of two interaction spaces: a physical and a virtual interaction spaces. The physical interaction space captures the current shape of the electronically-enhanced clay and sends its 3D shape over to the remote counterpart. Remotely located virtual interaction space displays the rendered 3D shape to the remote collaborators, and using a multi-user stereoscopic display or any other interactive display, and natural gesture tracking, allows the users to directly explore the 3D object from any angle, captures their manipulations of graphical and geometrical attributes and transfers them to the physical interaction space. On receiving the interactions, Couple-clay applies the manipulations to the physical object by overlaying projected images and, using a robot arm.

Coupled-clay makes two central contributions to the mixed-reality literature. The first contribution is related to extract 3D shape of the physical object. To do this, existing techniques use depth cameras [69] and laser scanners [77], however these techniques suffer from limited field of and hand occlusion, limiting their applicability in very dynamic situations. External markers can be used, but they require having a predefined shape. Our first contribution is to use embedded magnetic sensors [66] and a sensing table capable to extract 3D shape of the clay in real time. Given the magnetic nature of the tracking, the designer's hands do not create any occlusion problems and the tracking can be done unobtrusively in real-time. Our second contribution is the use of a robot arm and projection mapping in the physical interaction space aimed at reflecting the manipulations performed remotely in the virtual interaction space. The rest of this paper details Coupled-clay's implementation and discusses possible application scenarios.

5.2 Related Work

5.2.1 Interactive Multi-user Stereoscopic Displays

In terms of multi-user stereoscopic display, a display should present a pair of stereo-image with adequate disparity to each of the individual viewpoint of users if is aimed to be shared by multiple users. It always has to provide a distortion-free view to each of the users even if she moves around the display. Such display system must ensure that users can see the same virtual object at exactly the same location so that all the users can directly interact with it. Projection-based multi-user supporting stereo-displays (e.g., [88][181]) have a long tradition in

virtual reality field. Agrawala et al. [3] developed a system for enabling a two-user display to interact with a stereoscopic view of a shared virtual object, with each user standing around a table. Barco developed a two-user “virtual surgery table”, which provides stereo-images by differently polarizing the images of two active projectors [181]. Kulik et al. [82] introduced a stereoscopic wall-size display for six users, where each user has an individual viewpoint, by utilizing six different projectors each with time-sequential images. All these systems are limited to a small number of users and cannot be easily extended to support more users. IllusionHole [78], however, a multi-user stereoscopic tabletop display, consists of a display screen and a display mask with a hole in its center. With its simple configuration, this system provides stereoscopic images, which users can naturally reach out, and thus it is shared with others. By changing its various design parameters, the number of users could be easily increased. IllusionHole is also investigated in how the stereographic images are affected by measurements of the users’ eyes’ distances and positions errors [79]. However, tabletop stereoscopic displays may have huge components under the table, such as projectors and mirrors for bottom-projected displays (e.g., [78], [149]), requiring users to stand for an extended period of time with difficulty, impacting the naturalness of interaction between the users and the system.

5.2.2 Mid-air Interaction

By using direct mid-air interaction, high degrees-of-freedom manipulation can be achieved. Although this approach solves the difficulty of using a mouse and a keyboard, there is no consensus about the superiority of direct versus indirect multi-touch interaction. Sebastian et al.[80] focus on understanding directness on user performance for both 2D and 3D visualization cases. While directness reduces the task completion time, indirect interactions, such as an RTS (rotation, translation, and scaling) control widget, improve the precision of manipulation for 3D visualization. Martin et al.[50] combined direct RTS control widgets on a multi-touch table with a 3D stereoscopic display. This combination offers fast, easy interactions with rich stereoscopic content visualization. On the other hand, mid-air interaction tracking of a user’s fingers and hands provides natural interaction with 3D virtual environments. For example, Grossmann et al.[48] performed multi-user interactions in a volumetric display, where users can share the same volumetric content and manipulate it from their viewpoint. However, the imagery inside a volumetric display is enclosed by a protective transparent glass dome, preventing users from reaching in and grabbing the

objects. Holodesk[58] also provides natural interactions by allowing, though for a single user, to reach into a virtual environment to directly interact with a spatially rendered virtual environment just like grabbing a real object without using any head-worn hardware or input device.

5.2.3 Collaborative User Interfaces for Co-located and Remote 3D Interaction

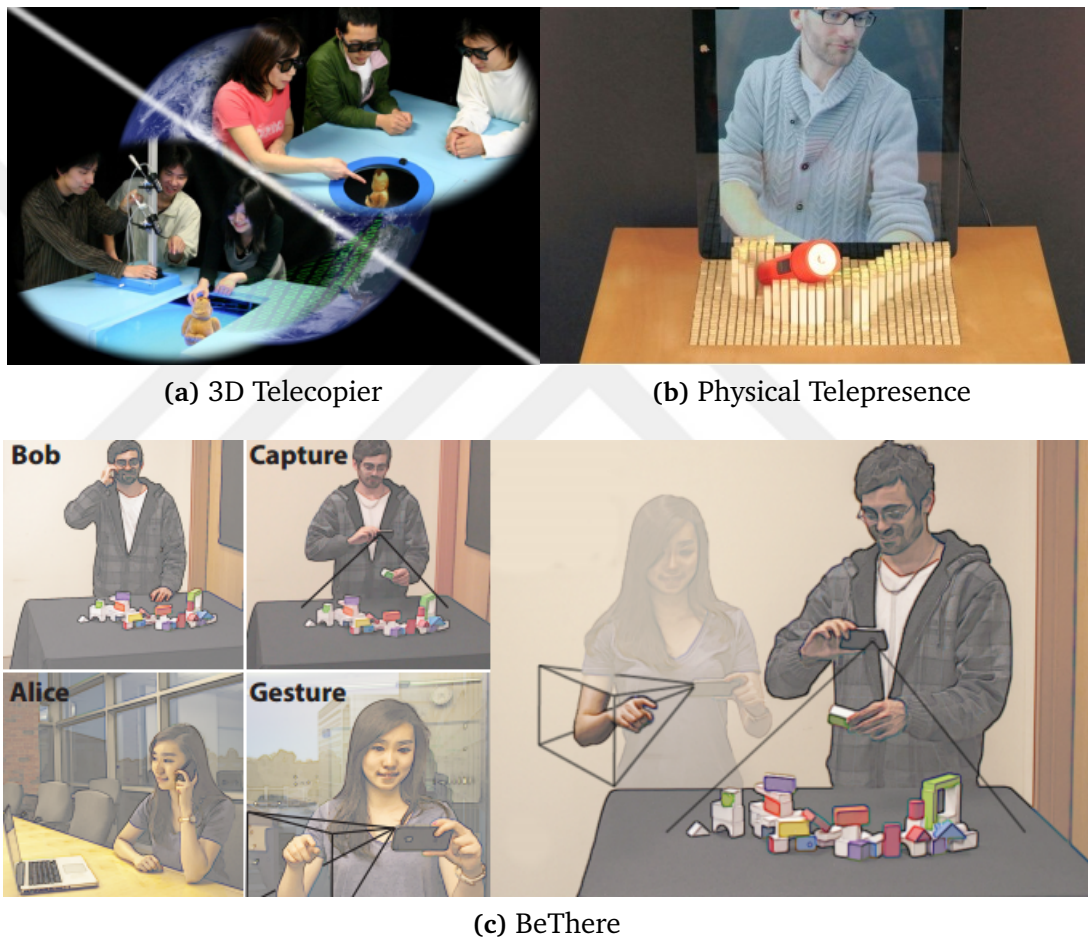


Fig. 5.1: Remote collaboration

Researchers work propose improving video-mediated collaboration by situating it in the physical world through the use of shared media spaces. Such kind of interactions mainly allow visually present remote participants on the screen. BeThere [154] is an example, which enables remote users to perform virtual interactions, such as pointing, in a local user's physical environment by utilizing a depth sensors to track the location of a user's fingers, as well as to capture the 3D shape of objects in front of the sensor (See Figure 5.1c). 3D

Telecopier [77] enables multiple groups of remote users share a 3D image of real objects and all the users can observe a 3D stereoscopic image from any direction (See upper right image in Figure 5.1a). However, the interface still lacks many physical aspects of collaboration that limits remote users' ability to collaborate through physical objects. In order to overcome this challenges, a remote telepresence robot have been proposed to enable remote workers to live and work with local coworkers as if they were physically there [85]. Leithinger et al. [89] proposed a method, Physical Telepresence (See Figure 5.1b), which based on shared workspaces with the ability to capture and remotely render the shapes of people and objects. They investigated how the representation of user's hands can be reformed to enlarge their capabilities for tele-operation.

5.3 An interactive multi-user 3D display using a 3D TV

5.3.1 Principle

The principle behind the IllusionHole is to place a mask with a hole on the stereoscopic tabletop display, and use a 3D tracking to detect the users' viewing positions around the table. This principle allows the users to see their own individual stereoscopic image pairs through the hole. The IllusionHole supports multiple dynamic users, who can freely move around the table during the observation. The image position on the display area for each user is computed according to his/her viewpoint measured by a 3D tracking sensors. Consequently, each user can observe the same stereoscopic image pairs (A, B, C) shown in an individual area. These pairs form a virtual object at the same position on the display from each of the viewpoints as shown in Figure 5.2. Each user around the table wears 3D glasses with tracking sensors to see stereoscopic objects through the mask hole. Moreover, since the display mask adequately occludes the image drawing areas of other users, they can only see their own corresponding image drawing areas.

5.3.2 Trade-off relationship among design parameters

There are some basic design parameters of the IllusionHole to determine its configurations: the size of the display surface ($h \times w$), display mask height (D), and the radius-es (major and minor axes, R_{major} and R_{minor}) of the elliptic

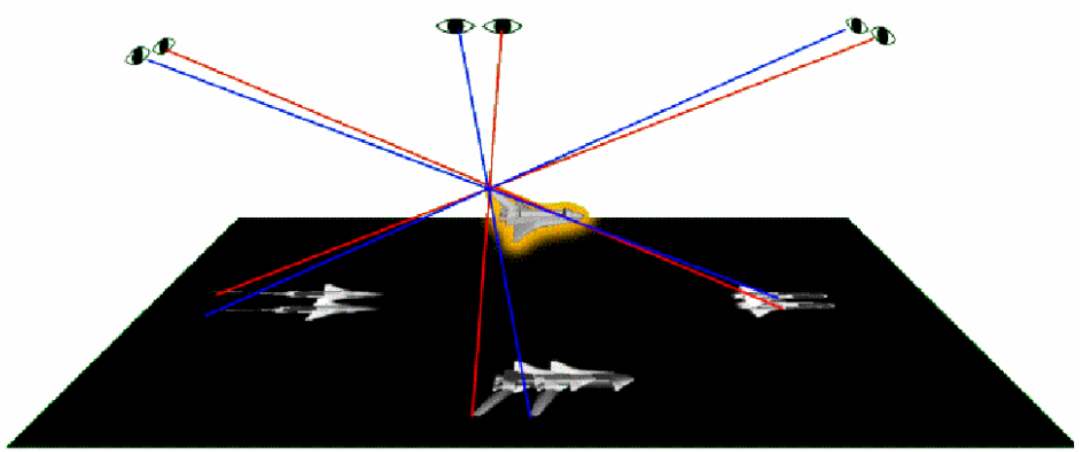


Fig. 5.2: Generation of stereo-images and image drawing area for each user in IllusionHole

mask hole (see Figure 3 (right)). They must be adjusted to satisfy the following conditions:

- Image drawing area for each user does not exceed the boundary of the TV screen.
- The image drawing area for each user does not overlap.

The valid height of a user's observable space is designed to be lower than the assumed height of the user's viewpoint. (In this study users are assumed to be seated.) The position of the image drawing area for each user is derived from two parameters, i.e., the user's viewpoint in the display coordinate system $(x_{eye}, y_{eye}, z_{eye})$ and the distance between the display surface and the mask D . The origin of the coordinate system is located at the center of display surface. By using this coordinate system shown left in Figure 5.4, the center of the image drawing area $(x_{center}, y_{center}, z_{center})$ is given by Equation (5.1).

$$\begin{aligned}
 x_{center} &= -x_{eye} \times \frac{D}{z_{eye} - D} \\
 y_{center} &= -y_{eye} \times \frac{D}{z_{eye} - D} \\
 z_{center} &= 0
 \end{aligned} \tag{5.1}$$

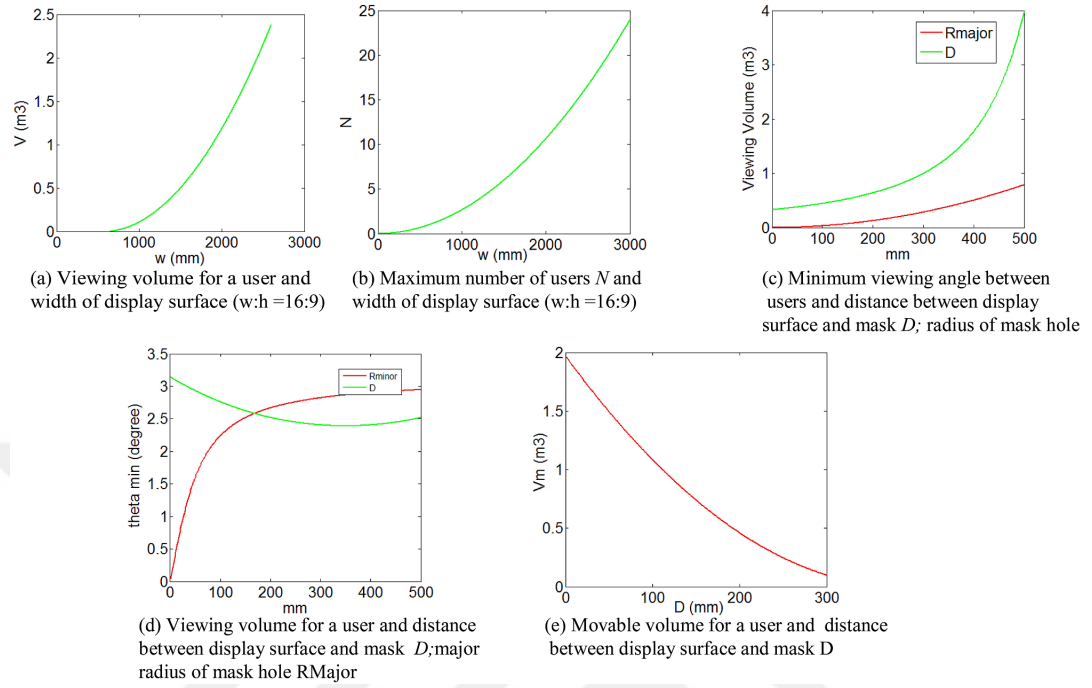


Fig. 5.3: Trade-off among parameters

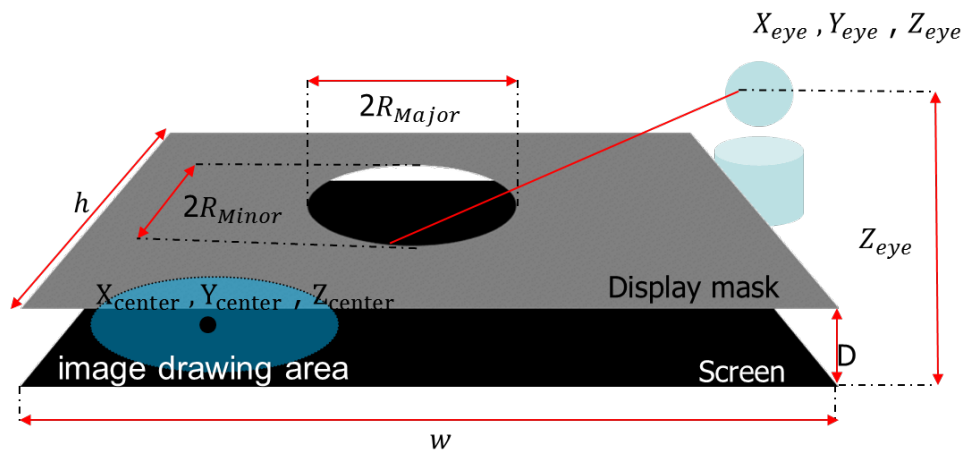


Fig. 5.4: Design parameters of the IllusionHole

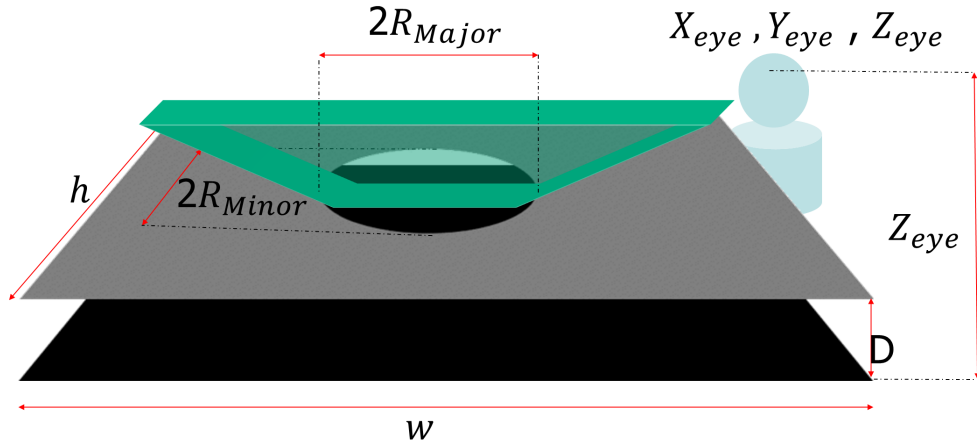


Fig. 5.5: Movable Volume for a User

$$\begin{aligned}
 r_{Minor} &= R_{Minor} \frac{z_{eye}}{z_{eye} - D} \\
 r_{Major} &= R_{Major} \frac{z_{eye}}{z_{eye} - D}
 \end{aligned} \tag{5.2}$$

5.3.3 Movable volume for a user

A larger movable volume provides a user with flexible observation of stereoscopic images because it allows the user to move around and change viewpoints in as large a space as the user wants (see Figure 5.5). Because the image drawing area must be inside of the display surface without exceeding its boundary, the minor diameter of the image drawing area $2r_{minor}$ is set so that it does not exceed the shorter side length of the display h ; therefore, $2r_{Minor} \leq h$ must be satisfied. From the relationship in Figure 5.4, the constraint on the height of a user's viewpoint is derived from equation (5.5) as shown by inequality (5.3).

$$z_{eye} \geq \frac{wD}{w - 2R_{major}} \tag{5.3}$$

This is then changed to inequality (5.4) by using equation (5.2) and inequality (5.3).

$$\begin{aligned} -\frac{h}{2} + r_{minor} &\leq x_{center} \leq \frac{h}{2} - r_{minor} \\ -\frac{w}{2} + r_{major} &\leq x_{center} \leq \frac{w}{2} - r_{major} \end{aligned} \quad (5.4)$$

For the constraint on the position of x_{center} and y_{center} , inequalities (5.4) are derived. The height of the user's viewpoint z_{eye} does not exceed the assumed user's viewpoint at z_{max} , so $z_{eye} \leq z_{max}$ must be satisfied.

$$\begin{aligned} z_{eye} &\geq \frac{(2|x_{eye}| + w).D}{w - 2R_{major}} \\ z_{eye} &\geq \frac{(2|y_{eye}| + h).D}{h - 2R_{major}} \end{aligned} \quad (5.5)$$

By using equation (5.2) and inequality (5.3), equation (5.4) is changed to equation (5.5). The movable volume for a user V_m where a user can observe an adequate stereoscopic image satisfying above conditions is given by Equation (5.6).

$$\frac{(w - 2R_{major})(h - 2R_{minor})}{3D^2} (z_{max} + \frac{D.w}{w - 2R_{major}})(z_x - \frac{D.h}{h - 2R_{minor}})^2 \quad (5.6)$$

5.3.4 Viewing Angle Between Users

When two users' points of view too closely approach each other's, their image drawing areas could over-lap. Therefore, the minimum angle between the users' viewing should avoid any overlap. Because the screen shape is rectangular, there are two angles that should be calculated (Figure 5.6). The minimum viewing angle between users should be adjusted if the number of users increases. The angle Θ_{users} , between two users' viewing lines, satisfies $\theta_{user} \geq 2\arctan(\frac{R_{major}}{D})$ and $\theta_{user} \geq 2\arctan(\frac{R_{minor}}{D})$, if the users are located at symmetric positions against the mask hole. Therefore, minimum angles given by inequalities (5.7) precisely can be obtained by using Equation (5.5).

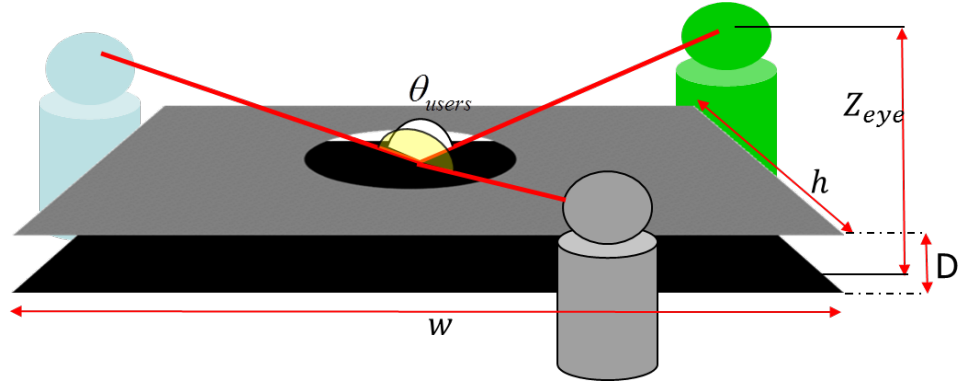


Fig. 5.6: Viewing Angle between Users

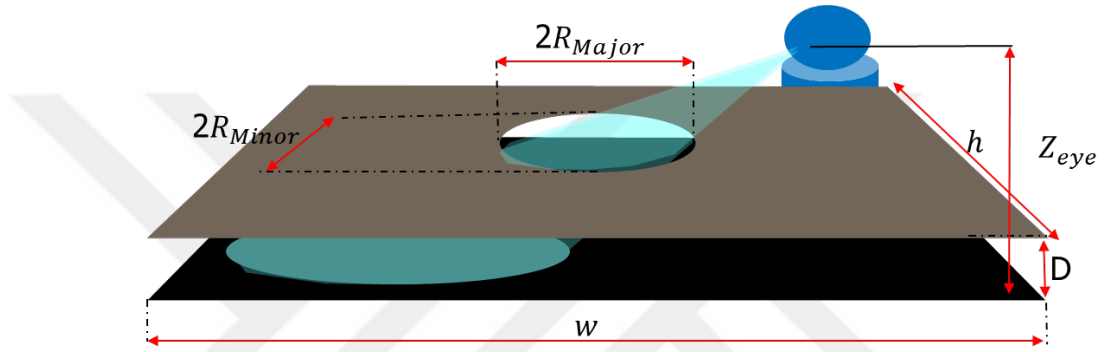


Fig. 5.7: Viewing Volume for a User

$$\begin{aligned}\theta_{user} &\geq 2\arctan\left(\frac{R_{major}z_{eye}}{D(z_{eye} - D)}\right) \\ \theta_{user} &\geq 2\arctan\left(\frac{R_{minor}z_{eye}}{D(z_{eye} - D)}\right)\end{aligned}\quad (5.7)$$

5.3.5 Viewing Volume for a User

Each user around the IllusionHole observes a stereoscopic image through the hole. Making the hole larger, within bounds, results in users observing larger stereoscopic images. Viewing volume for a user V_v is given by Equation (5.8) (See Figure 5.7).

$$V_v = \left(\frac{\pi R_{major} R_{minor} z_{eye}^3}{3(z_{eye} - D)}\right) \quad (5.8)$$

5.3.6 Maximum Number of Users

The image drawing areas for multiple users may overlap if the number of users increases and they stay too close to each other. The constraint to avoid overlaps must be analyzed according to the relative positions of the users' viewpoints and the positions of the image drawing areas, but the sum of all of the image drawing areas, n_{users} , for all the users must be smaller than the area of the TV's display as given by inequality (5.9), where the major and minor radii of the image drawing areas of all of the users are assumed to be equal to r_{major} and r_{minor} . The maximum number of users is derived from Equation (5.5) and inequality (5.9) as shown in inequality (5.10), where all users (n_{users}) are assumed to occupy the same image drawing areas. The maximum number of users N decreases according to the distance between the display surface and the mask D or as the radius of the mask hole increases.

$$\sum \pi \cdot r_{major} r_{minor} \leq h \cdot w \quad (5.9)$$

$$n_{users} \leq \frac{w \cdot h}{\pi \cdot R_{major} R_{minor}} \left(1 - \frac{D}{z_{eye}} \right) \quad (5.10)$$

5.4 Implementation Details

I use a 65-inch 3D Panasonic VIERA Plasma TV as the basic display system to make a prototype which is more tight enclosure compared to the previous IllusionHoles. A plasma TV with active stereo was favored over a passive stereo using polarization because users lose contrast when they tilt their head or shutter glasses while watching the TV. Since the IllusionHole needs to have a wide viewing angle to let users see continuously the displayed content from any directions, only a plasma TV provides such a wide angle. Configuration of the new IllusionHole including all dimensions is shown in Figure 5.9. A display mask with a hole at the center is designed to be in an elliptic shape (360 mm major axis and 210 mm minor axis lengths). In order to configure the TV to display side-by-side 3D, display resolution is set at 960 x 1080 (half of the Full HD). The maximum height of each user's head position is 1200 mm, the distance between the mask and the TV display is 230 mm, and the table's length and width are respectively 1006 mm, 1620 mm as it would fit the TV's size. Because the position of the image drawing area for each user is

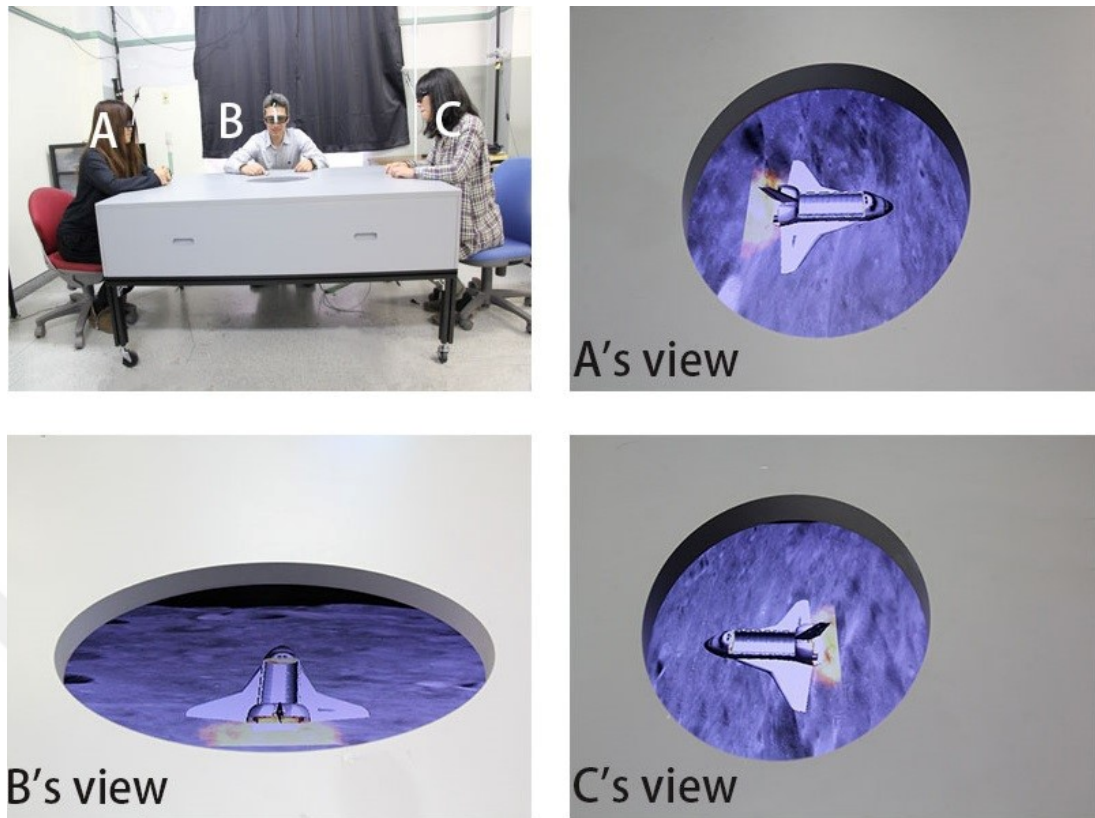


Fig. 5.8: Users observe the same virtual object from their own view-point. There is a 3D TV under the table.

dynamically calculated according to the users' eyes' positions by using an optical 3D tracking system (Optitrack), a virtual object is observed at exactly the same position in 3D space from all users. Users wear active shutter 3D glasses to observe stereoscopically displayed content. Graphics are created by a Windows 8 PC with NVidia Quadro FX 4800 graphics board. The IllusionHole's software creates side-by-side 3D signals that are supported by the 3D TV. When the TV receives the signals, it splits left and right frames to extract the frame for each eye. The TV then rescales these individual frames to a 960x1080 resolution. It then displays these frames alternately in a frame-sequential manner which works with integrated infrared emitter into the 3D TV on a corner of the display mask. The stereoscopic images are generated by using software written in C Sharp with Unity 3D IDE. Unity 3D is utilized to create high-resolution graphics. User positional data is streamed in real time to the system to compute relative position between a user and the image drawing area on the TV screen. The optical sensorial data is also processed in the Unity 3D editor.

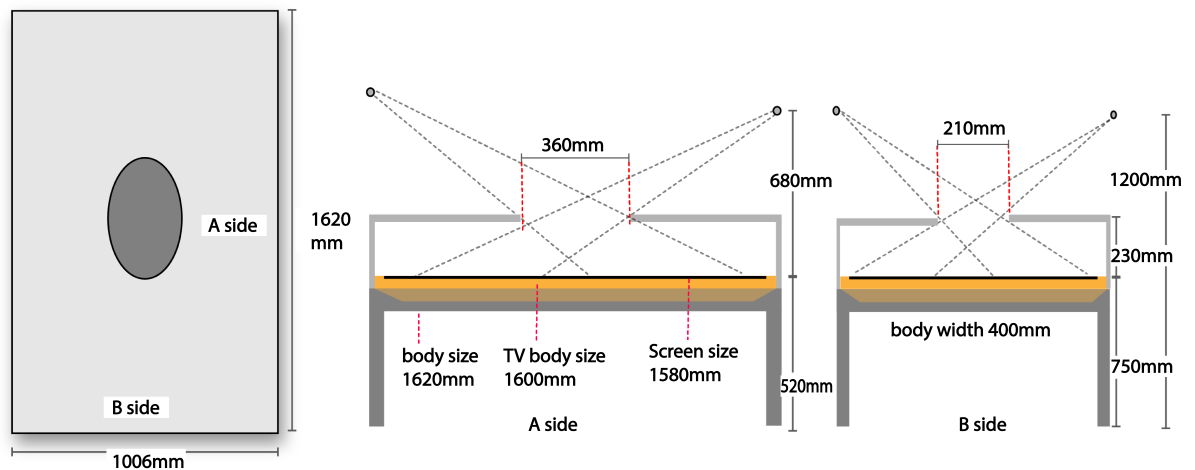


Fig. 5.9: Configuration of new IllusionHole design

5.5 Designing Direct Multi-finger Interface

I explore direct multi-finger object manipulation on 3D virtual objects based on the IH stereoscopic display. IH basically consists of a display and a display mask, which has a hole in its center. Each user can observe the stereoscopic image pairs shown in an individual area through the hole without seeing the image drawing areas of the other users (See Figure 1). Because IH's virtual space exactly corresponds to the physical space, the stereoscopic image pairs for each user around the table is at exactly the same position. The display mask hole allows these image pairs to be directly touched. Therefore, any manipulation of a virtual object can be performed and viewed by any of the users. Here, I implemented the direct multi-finger interaction technique (direct multi-finger method) to select, manipulate and release the objects.

5.5.1 Direct Multi-finger Object Manipulation

All gestures for manipulations are determined by optical markers on the tips of the fingers. The positional data of the fingers of the user are subsequently mapped to actions on the 3D virtual environment where the objects are presented. Figure 5.11(a) shows the markers fastened on user's fingertips for the direct multi-finger manipulation.

Selection and Release

The direct selection method is triggered when the position of the virtual object and the user's fingertips overlap. When one or two fingers touch the object, a

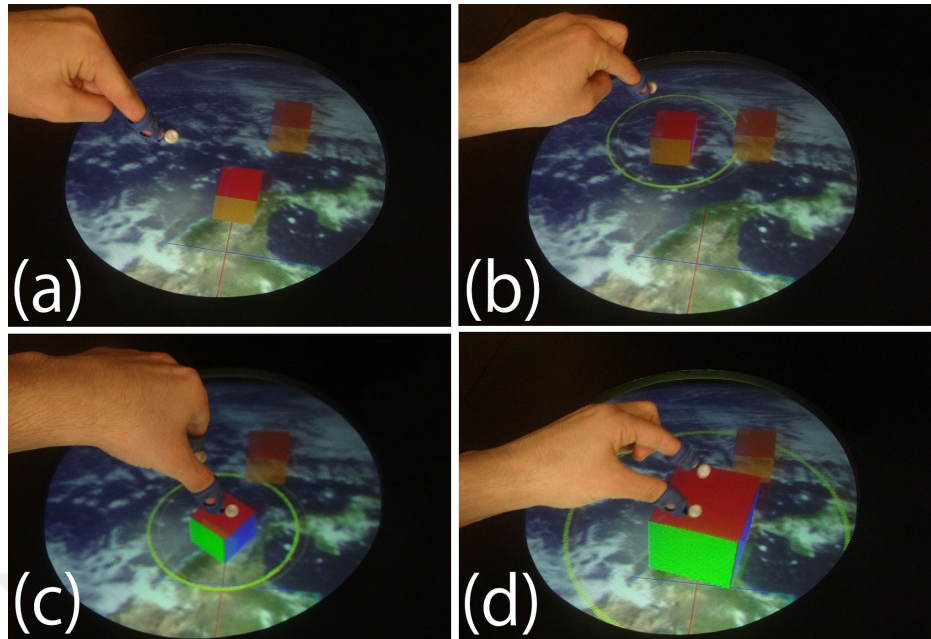


Fig. 5.10: Mid-air selection and manipulation tasks. Transparent box shows the final position, orientation, and scale of the interacting object (a) A hand is approaching to an object for selection. (b) Translation, (c) rotation, (d), and scaling

yellow circle horizontally appears around the object as visual feedback. When the user holds his fingers for a certain time (around 1.5 seconds), the visual feedback disappears, and then selection terminates. This time period is determined by pilot studies.

Translation

While an object is selected, 3-dimensional translation is performed either by moving one finger freely, or by equally translating two fingers simultaneously (See Figure 5.10.b).

Rotation

Rotation is initiated by touching the object with two fingers (only one axis of rotation about the upright z axis). Rotation is triggered when the user rotates his/her two fingers around the center of the object (See Figure 5.10.c). The gesture for rotation is imitated from common multi-touch interfaces with two fingers. During the rotation, translation and scaling can be performed simultaneously.

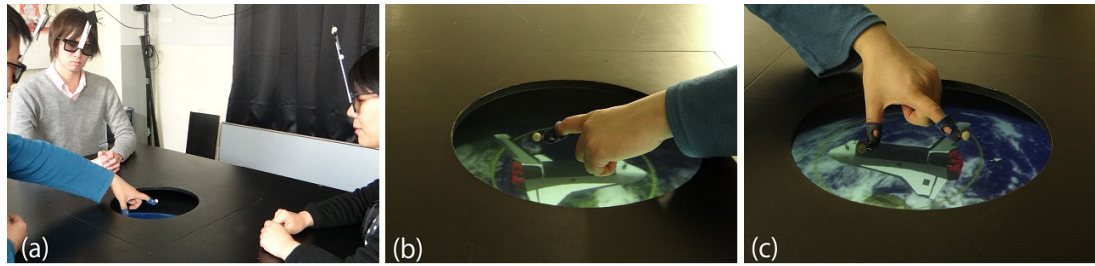


Fig. 5.11: Sharing direct object manipulation. A user directly interacts with an object and others look from their viewpoints (a). The appearance of the object from the user at the left side (b) and the user at right side (c).

Scaling

The size of a selected object is smoothly uniformly shrunk or extended by changing distance between tips of two fingers, i.e., closer together or further apart (See Figure 5.10.d). During the scaling, rotation and translation also can be performed.

5.5.2 Sharing Object Manipulation

Sharing of object manipulation is crucial for multi-user interaction, such as in collaborative 3D designing. The IllusionHole inherently allows multiple users to observe and directly point at 3D objects. By implementing multiple-fingers tracking, the users can collaboratively manipulate the objects. A user can manipulate the object while other participants sit on other sides of the table, observing the virtual object through the hole from their own point of view (See Figure 5.11). Here, the users are allowed to move around the IllusionHole during the observation.

5.5.3 Discussion

Design Simulation

Design parameters of the prototype of simulated IllusionHole configuration with four users are shown in Figure 8. When the distance between the display surface and the mask is too large, movable volume for each user becomes small, resulting image drawing area for four users easily exceed the boundary of the TV screen. However, when the distance between the mask and TV screen is too short, the movable volume for each user becomes large, resulting image drawing areas easily overlap each other. On the other side, when the radiuses of elliptic mask hole uniformly increase; viewing volume for each user becomes large,

resulting the image drawing areas easily exceed the boundary of TV screen. However, when the mask hole is too smaller, the viewing volumes become smaller, resulting image drawing areas do not easily overlap each other and stay in boundary of TV screen.

Implementation

The new IllusionHole is based on a tabletop metaphor, with each user sitting around a table. The size and shape of the new design allows more than four users. Using the new setup and bringing a 3D plasma TV, the new hardware and shape allows the users to observe, share and manipulate the same 3D content from their own viewing directions in a comfortable attitude, compared to the previous IllusionHole designs. It's apparently easy to load various 3D models for interactive manipulation with Unity 3D editor.

Practical Usage

Users observe stereo-image pairs through the hole at the center of the mask. Therefore, the system must keep image drawing areas inside the boundary of the TV screen for all the users seated around the table. To do this, movement ratio between the position of the drawing area and a user's head position must be decided for the users at the long side of the display. This is because the users seated at long sides of the table have narrower clearance of image drawing area due to utilizing the wide-screen (16:9) TV instead of 4:3 aspect ratio. It also easily allows users at the long side to manipulate virtual objects while also enabling users at short side to observe the manipulation.

5.6 User Study: Investigating Direct Mid-air Interface

In order to investigate its fundamental interactivity, I compared the direct multi-finger method (direct manipulation) with an indirect single-point method (manipulation at a distance). Twelve volunteers (8 males and 4 females) participated the experiment with ages between 22 and 34 (mean age 23). They had no previous experience in using any large stereoscopic display. The participant who performed the manipulation stood in front of the horizontal 60-inch diagonal IH [7] which uses passive stereoscopy and has the mask hole

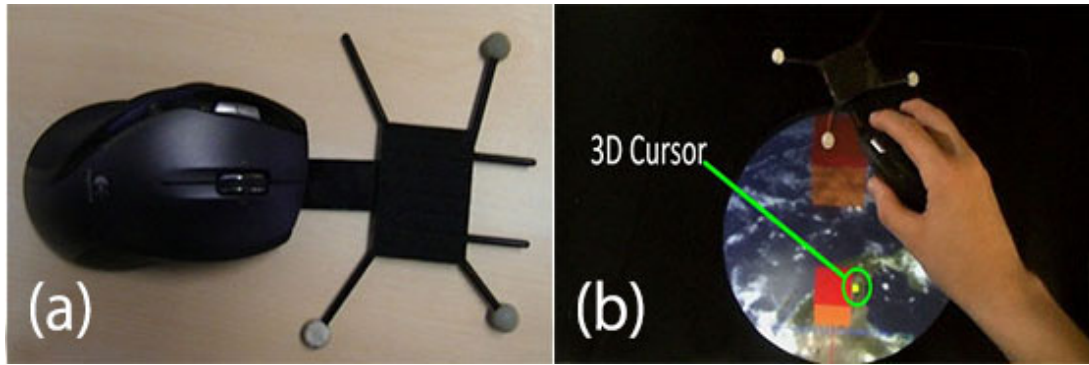


Fig. 5.12: 3D cursor: Indirect single-point technique (a) and its usage (b).

with diameter of 26 cm. Refresh rate of the screen is constant 60 Hz per eye of each user, even if the number of users increases.

5.6.1 3D Cursor

The reason of designing this competitor was that it could be a common interface with totally opposite concept to the proposed input method. For example, although directly touching a stereoscopic object with fingers is natural, it potentially has visual conflict [10]. On the other hand, using a cursor to perform distant object manipulation has fewer selection errors [8]; however, it has not offer natural interaction in 3D environments. This pros and cons must be discussed to see fundamental interactivity of the proposal interaction for object manipulation in a collaborative environment. Figure 4.a shows the indirect single point method which was designed based on a mouse button and a 3-dimensional single cursor with a 6DOFs tracker. A green cursor is displayed in 3D space (See Figure 4.b) and its movements correspond to movements of the 3D mouse. Selecting is triggered by merging the cursor inside the object. When the object is selected, a yellow circle appears. If the user holds the mouse for a short time, the yellow circle disappears. Translation is performed corresponding to 3D movement of the 3D mouse. A 1:1 movement ratio of method to cursor movement was used. To perform the rotation, the user rotates the 3D mouse about the upright z axis. The desired amount of rotation can be performed either at once, or in a few steps smoothly. Scaling is performed by rolling the mouse wheel as the common usage for zoom in/out, which is rolled forward to enlarge and backward to make the object small.

5.6.2 Experimental Design

The participants were asked to complete a 3D docking task implying five DOFs (x-y-z translation, rotation about the upright z-axis, and uniform scaling). In each task, two cubes (different colors on each side, see Figure 2) were displayed in the hole. One cube could be manipulated and the other was transparent indicating desired position, scale, and rotation. Task completion was based on users' sense of proportion for approximate difference between the transparently represented target and the manipulated object. The goal of the participants was to place the object at the transparent object's coordinates, to include proper rotation, translation and scaling. Here, I am interested in observing the difference on interaction time and errors between both methods for each user, rather than in their absolute values. Since all the participants used both methods, this would not change the differences between the results. On the other hand, if I used a threshold to define the matching between the target and the manipulated object, the task could be over even if the users were not satisfied (i.e. the manipulated object still did not seem to be in the right position), hindering our ability of measuring the real accuracy of the system. Position, size, and orientation of the target were changed between tasks to fit the size of the mask hole. Participants practiced two types of method for five minutes before the actual trials. A repeated-measures design was used with the within-participants. Independent variables were methods (direct multi-finger, indirect single-point), target size (object size x1.6 and x1.0), target distance (13 and 18 cm) and rotation angle (0, 90, and 180 degrees). The order of the two methods was counterbalanced among participants. All participants were shown the same set of target positions, angles, and sizes of the objects. Each participant completed 8 combinations of docking tasks with 4 repetitions, resulting in 64 trials in total. For each task, the system measured task completion time and errors in terms of position, scale, and rotation parameters. Participants were allowed to perform multiple selections and releases in a task. Task completion time was denoted as the total elapsed time between all selections and releases for each task. If a participant released the object by mistake, he/she could select it again to complete the task. At the end of the experiment, they answered a questionnaire based on a seven-point Likert scale. The entire experiment took approximately 50 minutes per participant.

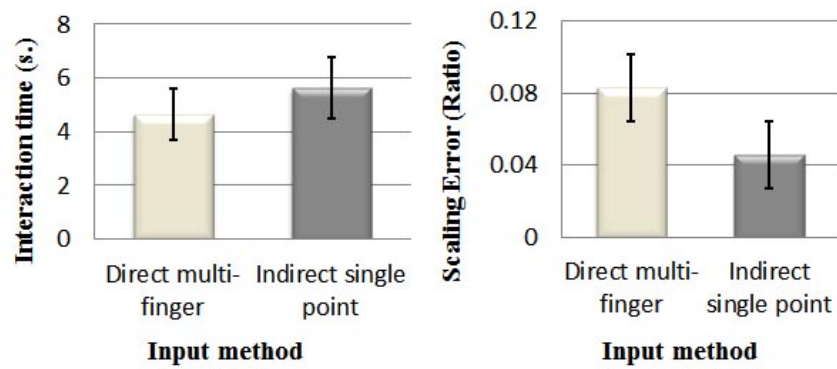


Fig. 5.13: Interaction time and scaling error

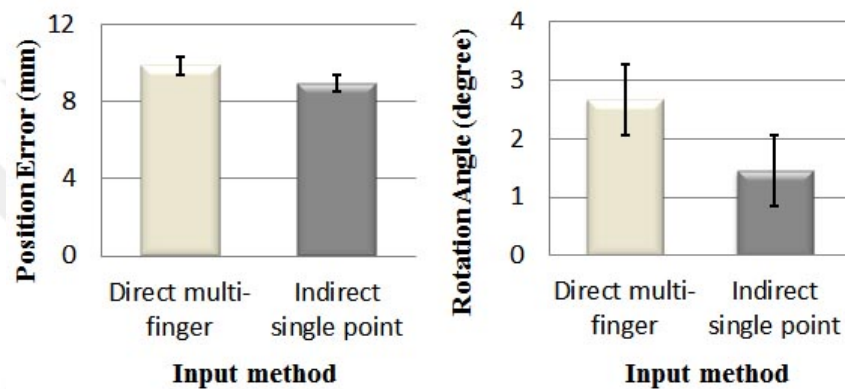


Fig. 5.14: Position error and scaling errors

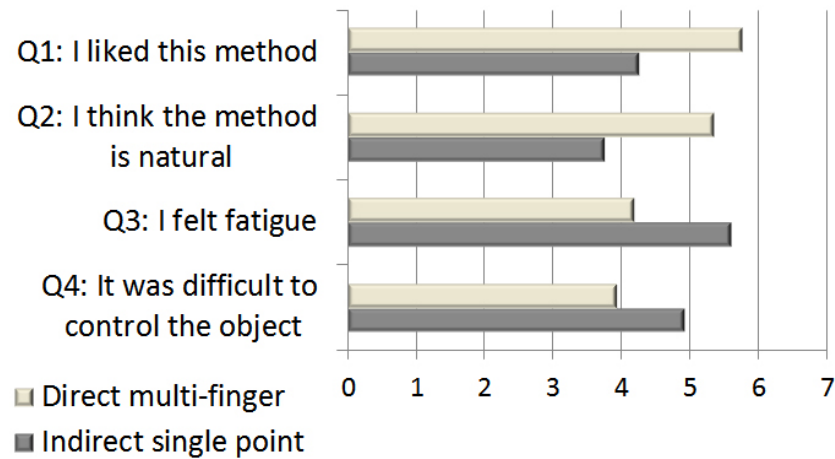


Fig. 5.15: General Feelings

5.6.3 Results and Discussion

A 2-sample t-Test was used to compare the two input methods because I was primarily interested in the comparison. Experimental results are shown in Figure 5 with error bars which indicate standard errors. In the task completion

time, direct multi-finger input was significantly faster than indirect single-point input ($t(119) = -2.56, p < 0.05$, see Figure 5.13 left). One reason may be that participants were familiar with the interface because multi-finger interaction is similar to multi-touch surface interaction. Moreover, the closely coupled visual and motor workspace on the whole of IH may have achieved greater performance. Although nearly all the participants could correctly target, scale and rotate the object with both inputs, they said that indirect single-point input felt more tiresome. Size, angle and position errors between the manipulated and target object for each input also were recorded. While there was a significant difference between the inputs in terms of scaling errors in favor of indirect single input ($t(119) = 3.05, p > 0.05$), see Figure 5.13 right), no significant difference was seen in terms of position errors ($t(119) = 0.749, p > 0.05$) and rotation errors ($t(119) = -0.179, p > 0.05$, see Figure 5.14). While there were quite small scaling errors, the reason was probably that scaling with a mouse wheel was more sensitive than the multi-finger pinch in/out gestures and because some jitter movements occurred when the participants scaled the object with fingers. Most of the participants reported that selection with the direct multi-finger method was notably easier. However, the markers attached to the fingertips were easily occluded while rotating the object.

Ratings of general feelings are shown in Figure 6. Ten participants preferred direct multi-finger input and gave many positive comments about it. They liked direct multi-finger manipulation with an average of 5.8 (Q1), which demonstrates satisfying overall docking tasks. The score for the naturalness is 5.3 (Q2). They also said that the direct multi-finger manipulation was less tiresome (Q3). Most of the participants pointed that the cursor-based selection with indirect single input was difficult (Q4). In both inputs, the participants were mainly able to manipulate the shown object, including selecting and releasing objects freely. However, the failure of position tracking arises from marker occlusions which disturb their natural interaction. The indirect single-point input has less occlusion error by about half. This is because the device was held up in the air during the interaction and had less tracking fails. However, this caused fatigue in the participants' hands. Additionally, most of the participants did not like the scaling the object with mouse wheel. Although selection with cursor is an easy method in 2D environments, it's apparently hard to select an object with 3D cursor in virtual space. Rather, when a user touches the object by finger, it can be easily selected. Directly touching a virtual object is easier than a 3D cursor-based interaction, especially for object selection. The reason

may be that the 3D objects are displayed within arm's reach, and since a 3D cursor is usually used for manipulation at a distance, it is not a natural way. In contrast, after the selection with direct multi-finger interaction, a small upward gap between the manipulated object and fingers seems acceptable due to the insufficient tracking accuracy and individual difference of eye disparity of each user. Moreover, if the displayed object is blocked by the manipulating finger(s) and hand, this will result in decreased visibility and interactivity. In other words, directly mapping between the finger movements and manipulated object is more important than directly touching the object. This allows users to observe the displayed 3D content from their own viewpoint. Since hands might block the view of the users who observing the manipulations, especially in case the manipulated object is small, indirect method could be useful. On the other hand, if the user scales and translates the object too much, stereoscopic image pairs in other users' individual area also scale and translate too much. This causes one user's individual image drawing area to be seen by other users. Using more accurate and higher DOFs direct methods would offer more natural and faster interaction. This is an interesting research topic and should be investigated further. Future work could include the investigation of the effect of higher DOFs, direct multi-finger object manipulations for multiple users, and more complex tasks that involve manipulations and sharing awareness at the same time.

5.7 3D Mid-Air Interaction using IllusionHole for Co-located and Remote Collaborations

Advanced tracking and display technologies create new opportunities for co-located and remote interactive collaboration. For instance, one of these new opportunities is sharing of physical objects, allowing both local and remote partners to modify their properties. For example, a skilled designer sculptures a 3D model of a new product, while a group of potential clients at a remote location interactively observe and interact with its rendered 3D virtual shape in real time. Clients see the changes made by a designer in real time, and their modifications are transferred to the physical object. With such cooperative capabilities, remote collaborators can directly influence the details of new designs during the design process as if they were collocated. For this purpose, I propose an interaction scenario for collaborative user interface using IllusionHole in this section.

Our 3D collaborative interactive system consists of two interaction spaces

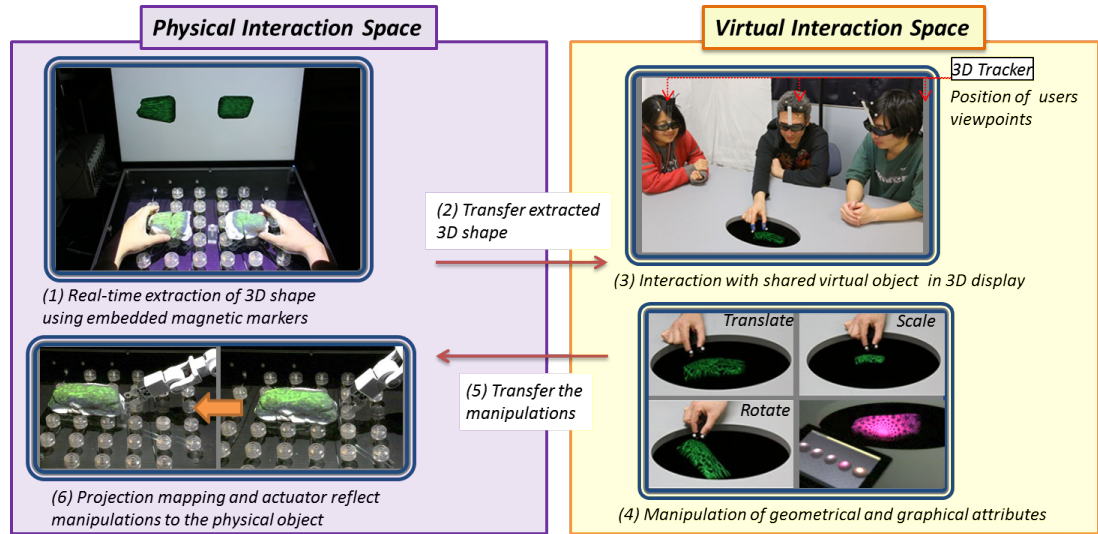


Fig. 5.16: Overview of bi-directional physical-Virtual 3D collaborative interaction with IllusionHole

connected each other through network: Physical Interaction Space, in which a user directly manipulates a physical object; and Virtual interaction Space, in which remote users observe and manipulate the rendered 3D shape of this object in real time. To accurately digitize 3D physical clay object and send it to the virtual interaction space, its precise 3D shape, including its position and orientation must be fully obtained even while the user are interacting with it by touching. This is a compelling challenge because hand/self-occlusion issues frequently occur when a vision based measurement system is used. The challenge in this remote space is to enable users to observe and interact with the virtual object which correspond to physical object. Another challenge on bi-directional interactions between the interaction spaces is achieved by reflecting the manipulations from virtual interaction space to the physical object. Figure 1 explains an example where a user in the physical interaction space manipulates a physical clay object, and remote users in virtual interaction space also interact with it through its rendered 3D virtual shape.

5.7.1 Virtual Interaction Space

This interaction space provides users distortion-free view from any viewpoints and manipulations of a virtual object. I use the IllusionHole with direct mid-air multi-finger interface [1], an interactive tabletop stereoscopic display; however, any other 3D displays with adequate interaction device can be used. The tabletop enables users to simultaneously observe 3D virtual object display's mask hole

so that they can share same explorations of the 3D object and provide explicit feedbacks to the user in the physical interaction space. Since the display's virtual space exactly maps on to the real space, the objects are exactly at the same position for each user. Therefore, tracking users' fingers enable them to touch and interact with them. A direct 3D interaction by multiple fingers is also performed to manipulate geometrical attributes (such as move, scale, and rotate) of an object. In addition, users can manipulate of graphical attributes of virtual objects.

Interactive 3D object modeling has highlighted the need for indistinctive, wireless and highly accurate 3D tracking system for small capture area (e.g., typical table-sized area, on tablet). Miller et al [109] proposed a method for 3D object modeling by clay lump that embeds several wireless AC magnetic markers. Due to the limited layout of the several markers, it achieved rough and large sized model, not in detailed. I developed an interactive clay modeling application that allows users to create more flexible virtual 3D objects through manipulation of actual clay using tiny and occlusion-free markers. Each of the 10 markers was covered with clay. Also, I put inside 1.5cm diameter sphere cases to keep the markers uniformly distribute inside the clay. The application uses the tracking data in the following way; assume each position of marker represents a sphere of clay, and check whether there are other clay spheres nearby. If so, connect the clay spheres with cylinder by each two, and with triangle by each three. Thus, a 3D mesh in the similar shape with the actual clay can be acquired by this application. Benefiting from the feature of occlusion-free, this application can basically model any kind of shape. Even when there is a hole or concave part inside the clay and user hand or fingers cover the clay, this application can still detect the model shape properly.

5.7.2 Physical Interaction Space

As Figure 5.16 shows, the physical interaction space consists of a magnetic tracking system that covers a physical interaction space, an overhead projector and a robot arm. The magnetic tracking system which used supports up to 10 identifiable, wireless, occlusion-free light and tiny markers, providing a semi sphere measurement space with radius of 150mm, and high position accuracy with average error smaller than 1.5mm. The tracking system is used to interactively digitize 3D shape of clay and there is no sense in any hand or finger occlusion. Each of the 10 markers is put into a small sphere plastic case, which is covered with clay object. Our prototype embeds a set of special occlusion-free

magnetic markers [66] into the object to dynamically extract its 3D shape. This 3D shape is rendered in the same interaction space for confirmation and also transferred to the remote space.

Magnetic Tracking System - Magnetic Tracking System [66], which used to fully track clay, uses LC resonating magnetic wireless markers, each has different id. The current tracking system supports up to 10 markers, providing a semi-sphere tracking space with radius of 150mm, and high position accuracy with average error smaller than 1.5mm. The tracking speed is 100 fps with one marker and 22 fps with 10 markers.

Arm actuator - A robot arm is used to reflect manipulations on the object in the virtual interaction space onto its corresponding object in the physical interaction space. Using a robot arm approach is promising to directly and dynamically convey interactions to the remote users in manipulating clay.

Dynamic Projection Mapping - Projecting the colored textures on the physical clay object is to change its appearance. Since I have precise extraction of 3D shape, in addition to its position and orientation, it is possible to produce a dynamically rendering of a colored textured using a projector while a user is interacting with electronic clay. A top-mounted projector is used to project a texture onto the clay.

5.7.3 Discussion

I imagine that the proposed system can be used for novice remote users to take part in any stage of designing 3D models, and reflect their expressions to a skilled designer, as an example. Moreover, it can be extended to use for various bi-directional interaction scenarios such as product promoter and remote clients, designer and examiners. Nevertheless, each workspace has different functionality; this can be altered according to role of users in a task. I plan to continually improve and evaluate the platform and to seek new capabilities. Additionally, advances in tracking system and display technology will make it possible to track, render and interact with more high resolution editable 3D objects. Thus, coupled-clay platform can researchers to integrate new emerging hardware as it becomes available.

5.8 Conclusion

In this chapter, I designed a new IllusionHole for collaborative multi-user mid-air selection and manipulation tasks as an example of flexible designs using a 65-inch 3D plasma TV. It allows four or more people to simultaneously observe individual stereoscopic image pairs from their own viewpoints in a more flexible way than previous IllusionHoles. The new prototype provides thinner, more compact, and makes the displayed content much easier to update thus, supporting a flexible collaborative workspace. I also designed a direct multi-finger interaction to perform fundamental manipulations of 3D objects (e.g., translation, rotation and scaling). It was evaluated and compared with a baseline cursor-based single pointing technique in a controlled study with 3D docking tasks. The study showed that direct object manipulation provides better user experience on a collaborative stereoscopic tabletop display environment. The study showed that direct object selection and manipulation facilitates 3D interaction in terms of performance (i.e., speed) and user preferences (i.e., task load index, overall preference) for 3D collaborative multi-user display.

Additionally, I know that for previous IllusionHole prototypes, users standing around the table have more freedom of movement during observation. However, standing scenarios give preference for short periods of use. The table might be designed to dynamically move up and down to allow the users take advantage of both seated and standing scenarios. For this reason the image drawing area for both scenarios should be considered as a future work. Therefore, the movement ratio between the position of the image drawing area and head position of users can be scaled when users change the scenario.

I also implemented a bi-directional interaction system by using IllusionHole that enables a user to directly manipulate a fully tracked physical object in mid-air and give access to remote users to view and interact with this object in real-time through its rendered virtual shape in 3D display at a virtual interaction space. This was achieved by utilizing embedded wireless sensors to extract 3D shape of a physical object in real time by which designer hands do not create any occlusion, and using of an actuator and projection mapping in the physical interaction space to reflect manipulations performed remotely in the virtual interaction space.

6

Effects on Guidance for 3D Mid-Air Interaction with a Public-exhibited Display

6.1 Introduction and Related Work

Public-exhibited displays takes part in our environment, and are visual forms of communication to understand, entertain, navigate and behave this environment. However, human-computer interfaces that utilize 3D mid-air gestures for these displays may be difficult to learn, owing to the fact that many of such interfaces lack the visibility of physical affordances that traditional interfaces can provide [124]. For example, a keyboard interface provides the visibility of buttons, thereby suggesting the physical affordance of button-based interaction. Similarly, a mouse is physically shaped to afford grasping, thereby suggesting to the user the affordance of hand-based manipulation. On the other hand, touch interaction is a common input for interaction with public displays. However, gestural-based mid-air interaction has a set of advantages over touch interaction as explained in [184]. Firstly, its appropriate for hygienic reasons since it does not require to touch a displays that everyone interact with. Secondly, users can remotely interact without come close to the display. Thirdly, it can make easy to perform expressive and highly visible gestures.

However, for 3D mid-air gesture interfaces such as the Leap Motion controller [193], such physical affordances are missing, leading to difficulties in learning how to use the system, also known as Learnability [136]. Furthermore this lack of affordance visibility may negatively impact the ability of the technology

to attract and sustain potential users in public spaces, such as a museum or exhibition. This ability to attract users and sustain attention is known as Recruitment [179], and can be measured through the study of Adherence [197], as observed when a person sustains attention on a task after first starting the task.

These problems may be solved by providing users with priming opportunities through Scaffolding Means [179], such as Feeding-back, Modeling and Instruction. Feeding-back refers to the provision of information regarding performance. Modeling refers to the process of offering behavior for imitation, such as the demonstration of particular skills. Instructing refers to a teacher telling a learner what to do. Unfortunately, it is still an open question as to how well such scaffolding performs for 3D mid-air gesture interfaces, which are different from traditional 2D interfaces [171]. Therefore, the contribution of this study is in determining which of the Scaffolding Means [179] (Feeding-back, Modeling or Instruction) are useful for increasing the Learnability and Adherence of 3D mid-air gesture interfaces facilitated by the Leap Motion Controller [193].

6.2 Methodology

The goal of this study was to collect a large amount of gesture data in different scaffolding conditions (Feeding-back, Modeling, Instructing), and compare the data from each scaffolding condition, so as to determine which scaffolding condition had the most benefit on Adherence and Learnability.

6.2.1 Using game Mechanics for Data Collection

To achieve this, I used an “in-the-wild” deployment technique, inspired by the work on WildAR [95]. WildAR is an experiment platform that makes use of games to enable movement-based experiments to be conducted on large populations, over relatively long deployment time periods, and it has been deployed in museums for the purposes of rapid and low cost collection of large movement datasets. In order to utilize WildAR-type protocols, I created a game (as defined as an activity with goals, rules, a feedback system, and voluntary participation [10] [71] [165]). I surveyed several different game genres that required physical movement, and decided to base our game on the rhythm and dance genre, made popular by a game called Dance Dance Revolution [7]. I chose this genre because the game concepts were simple, yet afforded a wide

range of design possibilities, while providing sufficient methods for controlling and constraining interactions. This made the genre suitable for experimentation work that required strict control over system parameters and variables. To play the game, the player used the Leap Motion Controller [95] as the sensor for detecting 3D mid-air hand gestures. To fulfill the requirement of having game goals, I had to create a scoring system based on recognizing gesture input. The problem was that typical gesture recognition systems required large training datasets tailored to a specific gesture detection method, and to the best of our knowledge, I was unable to find such a dataset available for use with the Leap Motion Controller. This necessitated the creation of a gesture detection system that did not rely on such a dataset of training data.

6.2.2 Defining an Atomic Gesture and Composite Gesture

To create such a gesture recognition system without the need for training data, I based my concept on the idea of an Atomic Gesture [12]. As stated in previous work, an Atomic Gesture can be defined as a gesture that cannot be further decomposed and which can be combined to create a larger composite gesture. For the purposes of this study, I define a unique Atomic Gesture as having the following properties:

- Can be created through physical movement
- Has a well-defined start and end state
- Has a unique signature that allows distinction from other Atomic Gestures
- Can be assigned meaning

My use of the Leap Motion Controller required a representation could be expressed via optical means, as the Leap Motion Controller is essentially a camera-based motion sensor used to detect movement in 3D space.

Therefore, from a visual point of view, an Atomic Gesture can be represented by a straight line with a length and a direction. To make the line atomic, its length has to be short enough to be included in all possible composite gestures, yet long enough to be detectable by the Leap Motion Controller. When combined with a distinct direction in 3D space, the Atomic Gesture forms a unique signature. A Composite Gesture can then be formed by combining Atomic Gestures together. As expressed in mathematical form, an Atomic Gesture (G_a) and Composite Gesture (G_c) as follows:

$$G_c = \sum_{i=1}^n G_{ai}$$

where $G_{ai} = G_a(\text{length}_i, \text{direction}_i)$.

Based on this representation, it becomes possible to capture and store detected movements in a relational database of gestures using the Leap Motion Controller.

6.2.3 Representing Gesture Signatures in a Relational Database

As defined in the previous section, I can represent an Atomic Gesture as having length and direction, and a Composite Gesture as a combination of Atomic Gestures. When this definition is implemented as coordinates in 3D space represented by the XY and XZ planes, it becomes possible to plot (in relation to an arbitrary origin) the start and end points of an Atomic Gesture, in a chronological sequence, so that a Composite Gesture is formed (See Figure 6.1). Mathematically, this can be represented as:

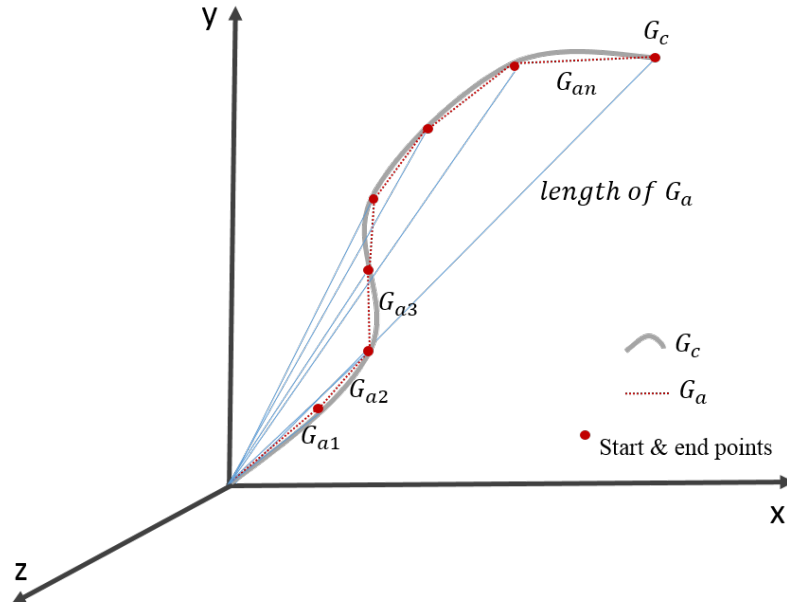


Fig. 6.1: Graphically, a Composite Gesture in 3D space can be represented as a combination of Atomic Gestures (straight lines, with lengths and directions, with respect to an origin). The Atomic Gestures can be arbitrary short, as long as they can be detected by a suitable detector.

$$\theta_{xyi} = \arctan\left(\frac{y_{Gai} - y_{Ga1}}{x_{Gai} - x_{Ga1}}\right)$$

and

$$\theta_{xzi} = \arctan\left(\frac{z_{Gai} - z_{Ga1}}{x_{Gai} - x_{Ga1}}\right)$$

where $G_{a1} = G_a(\text{length}_1, \text{direction}_1)$ which is the first Atomic Gesture in the list.

Based on this representation, I only need to store the direction (in terms of angles (θ_{xyi} and θ_{xzi} , for all i) with respect to the origin) and the length (in terms of the distance from the origin) of each Atomic Gesture, and store multiple Atomic Gestures in chronological sequence of the Composite Gesture's creation. This stored information then becomes the gesture signature, which can then be used to create a gesture recognition system.

6.2.4 Creating a Gesture Recognition System

Once the Composite Gesture can be represented in a relational database as Atomic Gesture signatures consisting of direction and length, it becomes possible to create a recognition system by capturing a physical gesture using the Leap Motion Controller, sampling the gesture according to a desired sampling rate, extracting the direction and length information from each sample, and directly comparing this data with each Atomic Gesture signature in the database, so as to recognize the Atomic Gestures, and by association, the corresponding Composite Gesture. This approach enables parallel processing of the gesture information, leading to fast recognition capability.

6.2.5 Designing the Game Concept and Scoring System

The game was created, known as Gesture Gesture Revolution (GGR) gave players the goal of obtaining the highest possible game score. To obtain a high score, the player had to perform certain gestures when instructed to do so by the game. The list of gestures was finite and was displayed according to a fixed order. If the player performed the correct gestures at the correct time, the player would score points, shown as an increasing score displayed onscreen, otherwise the player would lose "life", as shown as a decreasing percentage onscreen from 100% to 0%. The player would complete the game if s/he performed enough gestures correctly to prevent his/her "life" from dropping to 0% before all the gestures on the list were displayed. Players would then compete with

one another for the highest score possible. The scoring system therefore used the gesture recognition system described in Section 2.4 to compare the player's gestures with the gesture signatures in the database to determine the player's score.

6.3 System Implementation and Deployment

The system was implemented using a client-server architecture consisting of a database server backend, and a game client frontend.

6.3.1 System Architecture

Figure 6.2 shows the basic system components. The server hosted the game software (created in Unity [175]), and the MySQL database (implemented as part of the WAMP [18] server package). I used a desktop computer with an Intel i7-6700 CPU and 8GB of RAM, running Windows Server 2012 R2 to house the system. The Leap Motion Controller acted as the game controller, as the game required players to perform mid-air hand gestures to play the game. The large display acted as the visual interface for the game.

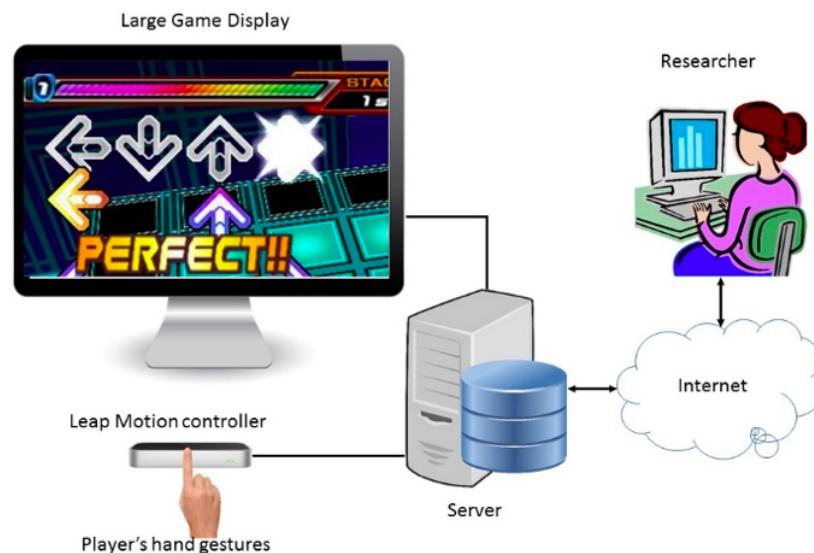


Fig. 6.2: Client server architecture, operating within the cloud, enabling researchers to collect data and change system parameters remotely.

6.3.2 Game Design

For GGR, I selected four gestures to be used: swipe up, swipe down, swipe left, and swipe right. When the game starts, an arrow appears at the top of the screen, and starts moving downwards (Figure 6.3 (a)). The arrow indicates to the player which gesture is to be performed. When the arrow reaches the bottom of the screen, the player must perform the correct gesture in order to score points. The player scores a different amount of points, depending on how close s/he is to the bottom of the screen before s/he completes the gesture. The closer s/he is to the bottom of the screen, the more points s/he receives. However, the player receives no points if s/he performs the gesture too early (too far from the bottom of the screen), or too late (the arrow moves out of the screen). In Unity, this scoring mechanism was implemented using invisible box colliders. Figure 6.4 illustrates how the colliders were positioned in the game level to create different scoring zones. If the player performed the correct gesture in the blue zone, s/he would receive 10 points. If s/he performed the correct gesture in the magenta zone, s/he would receive 50 points. If s/he performed the correct gesture in the yellow zone, s/he would receive 100 points. If a gesture was not performed correctly, or it was performed outside of these zones, or not performed at all, the player's "life" would decrease by 10%, and no points would be gained by the player. The sequence of gestures to be performed in the level was predetermined in the database, and the game simply loaded the level from the database whenever the game started. This enabled to strictly control the parameters of the game level, and yet be agile enough to change game level whenever the experiment required it.

6.3.3 Deployment Venue Design

The system was designed to perform like a typical museum exhibit. As shown in Figure 6.5, the system consisted of a Panasonic TH-P50ZT5 50-inch display mounted on a Sanwa display stand. The display was connected to a desktop PC secured and locked to the base of the stand.

As the stand had a small table attached to it, an acrylic box was attached to the table with strong 3M double-sided tape, and covered with a black cloth for aesthetic reasons. The Leap Motion Controller was placed on the top of the acrylic box, and was secured to the box using strong 3M double-sided tape, through a hole cut into the black cloth. This made the system robust to everyday usage, as the entire system was now a single unit that could

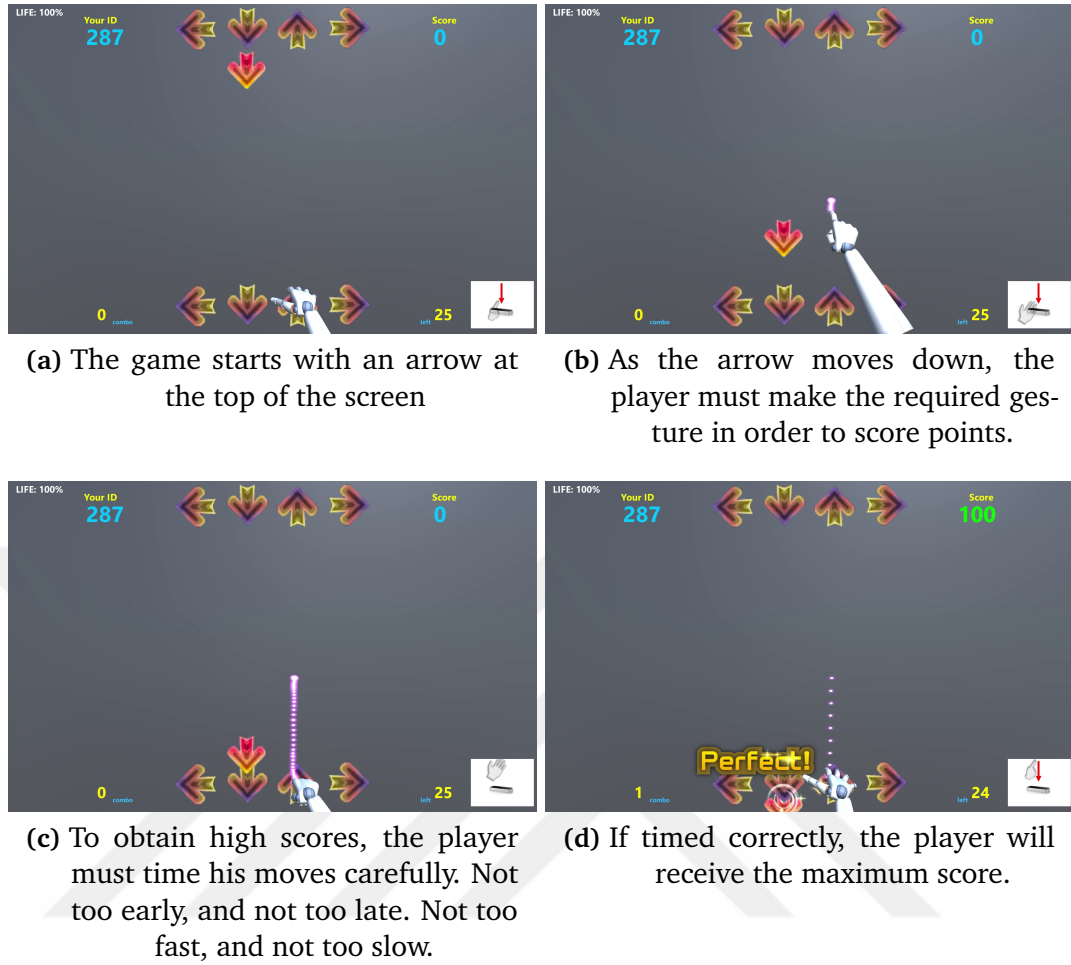


Fig. 6.3: Illustrated sequence of game dynamics

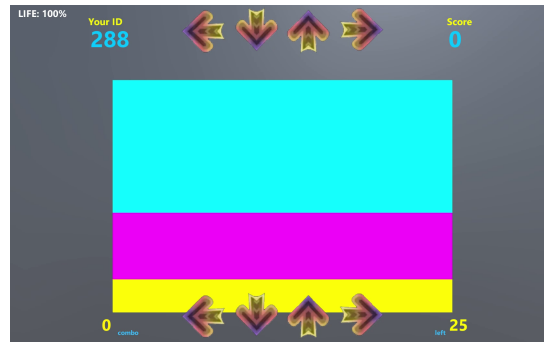


Fig. 6.4: The areas for scoring are implemented as colliders in Unity. The blue region gives 10 points, the magenta region gives 50 points, and the yellow region gives 100 points. During gameplay, these regions are not visible, so the player needs to learn when and how to gesture to score the highest points.

withstand accidental bumps and hits from visitors. The system was configured to automatically power-on and launch the required software, everyday at 08:30.

The system automatically shut itself down at 20:30, as per requirements of

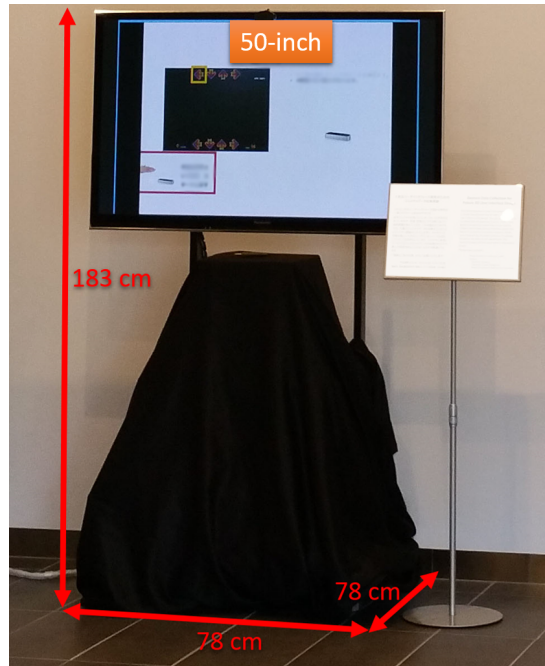


Fig. 6.5: Photograph of the game system made into an exhibit and deployed at the ground floor within a university campus. The dimensions of the system were carefully designed to ensure ease of access and use.



Fig. 6.6: The exhibit was deployed next to the lift lobby, ensuring a high participation potential.

the exhibition venue sponsors. The exhibit was deployed at the ground floor of the Research Institute of Electrical Communication, Tohoku University. Based on our observations, on the average, about 400 individuals passed through the venue per day, and about 25% were female. I conducted an analysis of human traffic flow at the site, to determine the placement of the exhibit. As shown in Figure 6.6, the deployment site was located right next to the lift lobby, and about 80% of human traffic moving in and out of the building had to pass by the exhibit, thereby increasing its chances of being played with. Figure 6.7 shows a

group of passers-by playing the game.



Fig. 6.7: Passers-by playing the game.

6.4 Experiment Design

The system has been deployed the exhibit from January 2016 to March 2016. During this period, an experiment was conducted to measure the effect of different guidances (Scaffolding Means) on learning and adherence mid-air gestures.

6.4.1 Independent Variables

For this experiment, the independent variable was that of Scaffolding Means. There were three levels and one control, thereby forming four conditions:

- *Modeling*: In this condition, a two-second animated video accompanied each new gesture in the game. For example, when the “up” gesture was required for the first time, an animation demonstrating what movements the player should make was shown. The same happened for the “down”, “left” and “right” gestures. Note that the demo video only played once for each new gesture, and the videos did not appear for subsequent gestures of the same type. Figure 6.8 (2.1-2.4) shows a multi-frame illustration of this.

- *Feeding-back*: In this condition, the player was kept informed of the game's hand "recording-state", which meant that the players were kept informed of when their hands were making movements that the system recognized as gestures. For example, when the system detected that the hand was in an idle non-playing state, a blue icon was shown on the screen (Figure 6.8(3.1) and Figure 6.8(3.4)). When the system detected that the player's hand was in a playing state, a red icon was shown (Figure 6.8 (3.2) and 6.8 (3.3)). In this way, the player knew exactly when the system was detecting his/her gestures, and when it was not. In this way, the player could adjust his/her gestures to better cater to the system's detection capabilities.
- *Instructing*: This condition was similar to that of Modeling, but instead of an animated video, a pictorial representation in the form of a sequence of instructional images was shown to the players.
- *Baseline/Control*: The basic system was the same for all the other three conditions as stated above, excluding their individual features of Feeding-back, Modeling and Instructing. Figure 6.8(4.1-4.3) shows an illustration of a baseline system without Feeding-back, Modeling and Instructing.

Each condition was deployed for one week from Monday 0830hrs to Friday 2030hrs, after which the next condition was deployed into the system. The sequence was:

- Week 1: Baseline,
- Week 2: Modeling,
- Week 3: Feeding-back
- Week 4: Instructing

At the end of Week 4, the system was taken offline.

6.4.2 Dependent Variables

The dependent variables were that of Adherence and Learnability, parameterized into specific measurements:

- Adherence was measured by counting the number of gestures performed by the player, divided by the total number of gestures required per game

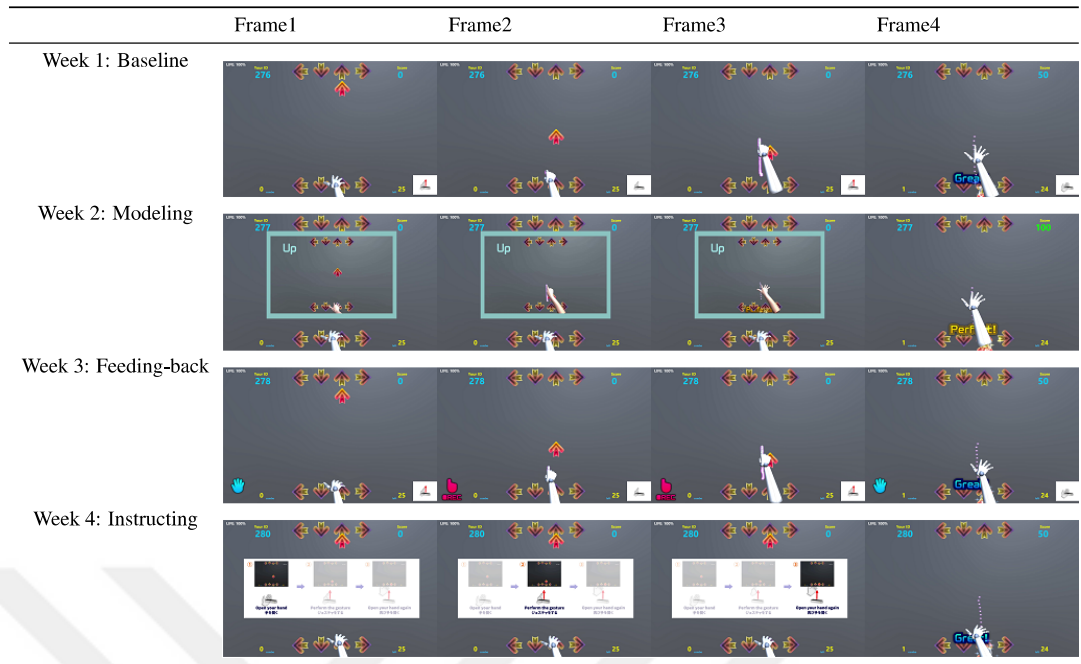


Fig. 6.8: Experiment conditions from Week 1 to Week 4. The Baseline condition was what the other conditions built upon, meaning that all conditions included features of the Baseline condition. Conditions from Week 2,3 and 4 were then implemented exclusively for that week only, meaning that there were no feature overlaps between Weeks 2, 3 and 4, other than what was already available in the Baseline condition.

stage (as defined in the game stage database). This would give a percentage of game participation, with a higher percentage meaning that more players were playing more of the game (and adhering to the task) before giving up.

- Learnability was measured through two parameters:
 - *Correctness*: This was measured by counting the number of correct gestures made by the player, divided by the total number of gestures per game stage. This would give a percentage of how well the player was playing, with a higher percentage meaning that the player was playing the game more correctly than a player with a lower percentage.
 - *Score*: This was measured by summing up the player's total score. A higher score would mean that the player had learned to play the game better than a player with a lower score.

6.4.3 Experimental Protocol

The experiment protocol is as follows:

1. A potential player walks into the venue
2. His/her attention is drawn to the large display, which shows an animated sequence of instructions on how to play the game, along with a disclaimer and statement of research intention. This animation plays on a loop until the game starts.
3. If the user decides to play the game, they hover their hand over the Leap Motion Controller for about 2s to activate the game.
4. The system detects the player's hand, and commences starting procedures.
5. The system asks the player to specify their profile number or age and gender using mid-air hand movements to move a virtual cursor to select the appropriate response.
6. Once done, the system shows the player an animated gameplay tutorial played on loop.
7. When the player decides that s/he is ready, s/he hover's his/her hand over the "Play" button.
8. The game starts.
9. After the game ends, the system displays the player's score on a leaderboard.
10. The system asks the player to rate his/her enjoyment of the game on a scale of 1 to 5.
11. Once done, the system generates a user profile number and shows it to the player. The player can then use the profile number the next time s/he plays the game. This is so that the system can keep track of repeat players, and so that players can keep track of their own progress.
12. Finally, the system returns to the initial animated sequence, awaiting the next player activation.

6.4.4 Results

During the four-week experiment period, the system recorded a total of 189 unique players (repeated plays have been removed) (Mean age=26.9, SD=9.54, 42 females). The number of users played the game per each week is shown in the Table 6.1. Based on the data, I conducted statistical analysis in terms of Adherence, Correctness and Score. I collected the data from repeat-players, meaning that all the data was from players who only played for the first-time, enabling me to conduct a between-subjects analysis.

Tab. 6.1: Number of users played per week

number of users	weeks
32	1
45	2
25	3
20	4

Adherence

As shown in Figure 6.9, Bonferroni-corrected Chi-squared comparisons revealed that players in Week 2 and Week 3 produced significantly more gestures than in Week 1 (Week 1 vs. Week 2: $\chi^2 = 51.3$, $p < 0.001$, Week 1 vs. Week 3: $\chi^2 = 33.6$, $p < 0.001$).

There was no significant difference between Week 1 and Week 4 ($\chi^2 = 4.29$, $p = 0.0384$).

Correctness

As shown in Figure 6.10, Bonferroni-corrected Chi-squared comparisons revealed that players in Week 2 and Week 3 produced significantly more correct gestures than players in Week 1 (Week 1 vs. Week 2: $\chi^2 = 32.6$, $p < 0.001$, Week 1 vs. Week 3: $\chi^2 = 10.2$, $p = 0.0014$).

There was no significant difference between Week 1 and Week 4 ($\chi^2 = 0.00$, $p = 1.00$).

Score

As shown in Figure 6.11, an ANOVA test and a post-hoc Tukey test revealed that there was a significant effect between Week 2 and the other weeks ($F[3,189]$,

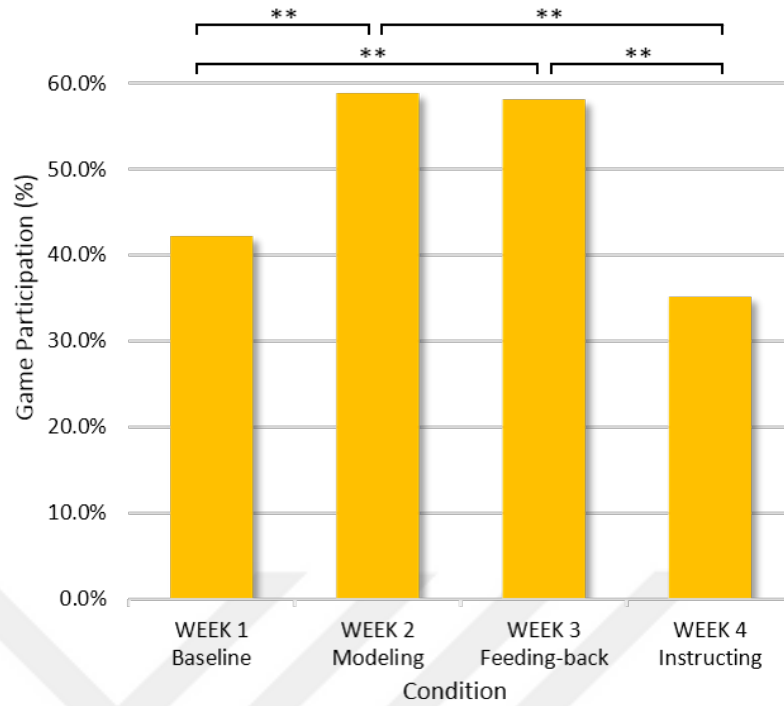


Fig. 6.9: Graph of Adherence vs. Condition. ** depicts $p < 0.01$.

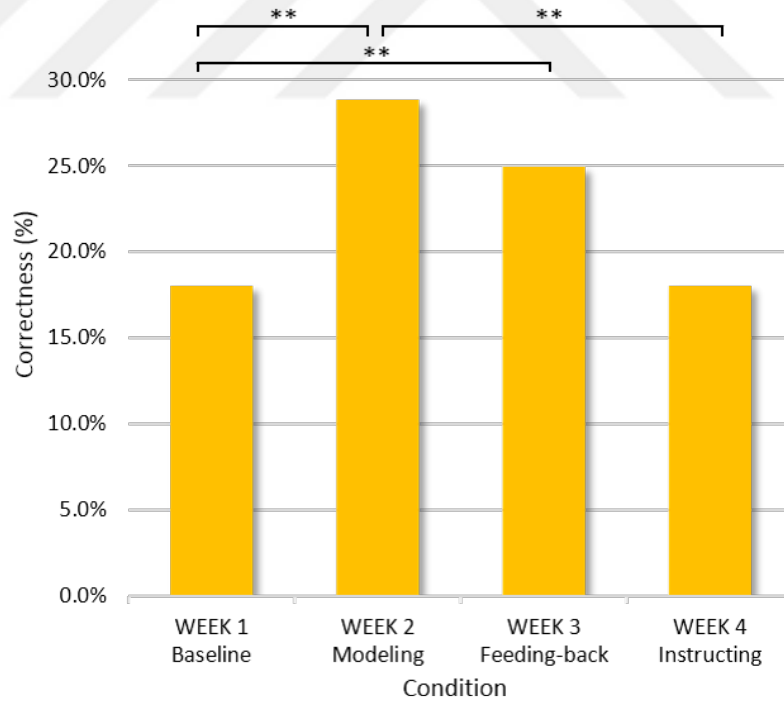


Fig. 6.10: Graph of Correctness vs. Condition.

$p < 0.001$). Specifically, Week 2 was significantly different from Week 1 and 4, but not from Week 3.

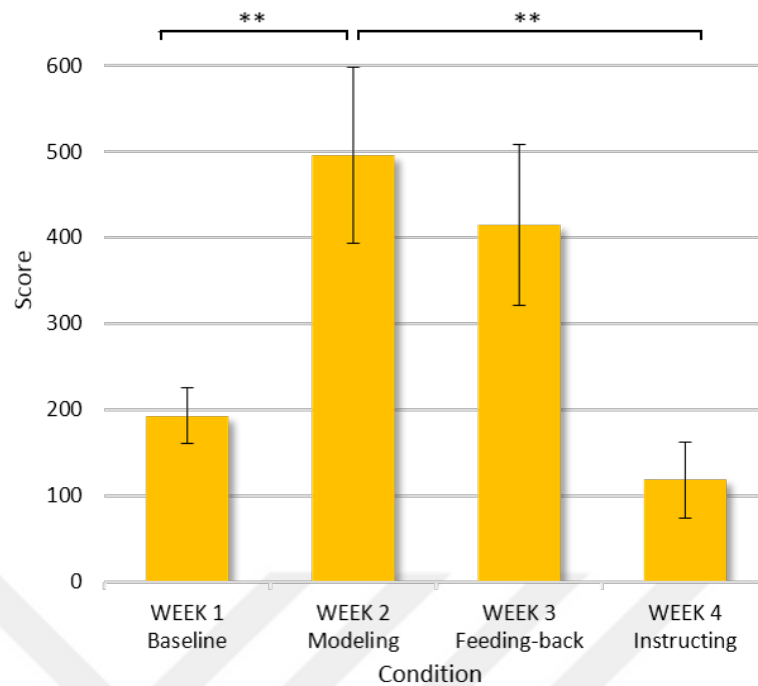


Fig. 6.11: Graph of Score vs. Condition. Error bars depict standard error.

6.4.5 Discussion

Pertaining to the discussion about Adherence, with the results shown in Figure 6.9, I can clearly see that Adherence was significantly improved through Modeling and Feeding-back, but not through Instructing. Similarly, Figure 6.10 shows that Modeling and Feeding-back were also more effective at increasing Correctness than the baseline, but not Instructing. However, the same cannot be said of the Score, because only Modeling produced a result that was significantly better than the baseline (Figure 6.11). From these results, I can rank the Scaffolding Means in terms of their effectiveness in improving Adherence and Learnability. Modeling was clearly the best because it significantly improved all of the dependent variables over the baseline. Feeding-back was the next best choice, as it was effective at improving Adherence and Correctness, but not Score, meaning that it was only partly effective. Instructing was clearly not effective at improving any of the dependent variables.

These findings enable to think over why certain Scaffolding Means were more effective than others in improving Learnability and Adherence. I begin with a discussion on Learnability, and start with a comparison between Modeling and Instructing. As demonstrated earlier in the study, Modeling and Instructing were very similar in implementation, just that one was an animated video

illustration, and the other was a sequence of still images. This suggests that continuous animation or movement was important in facilitating learning of the game, as it gave a clearer understanding to the player about what to do, by providing clear directions to be imitated. Instructing may have been less effective, because although players were told what to do verbally and pictorial, they still had to “fill in the blanks” in terms of how actually to do it, making the game harder to learn.

Feeding-back “fills in the blanks” in a different way. In our implementation, players were given feedback on whether the system was detecting their hand gestures. Therefore, even though they were not told exactly what to do, players could figure out through trial and error on how best to play the game. This would explain the comparable performance between Modeling and Feeding-back in terms of Correctness. However, these two methods differed in their Score, with Modeling producing higher scores than Feeding-back. To understand why, I must first understand the complexity of the game scoring system, as compared to the complexity of the gesture detection system. Players in the Modeling condition were explicitly shown that they should time their gestures at the exact right moment to score the maximum points, while players in the Feeding-back condition were not informed in the same way. Therefore, again, players in the Feeding-back condition had to resort to trial and error, and therefore taking longer to learn the scoring system. Hence, their scores were lower than players in the Modeling condition. Having three levels of scoring (10, 50 and 100 points), the scoring system was more complex than the gesture detection system that had only two levels (detected / not detected), therefore Feeding-back was less efficient at facilitating learning of more complex mechanisms than Modeling.

Still, players in the Modeling and Feeding-back conditions were much more successful in the game than their Instructing counterparts, and this had an effect on Adherence. This could be due to several reasons. First, when feedback was provided, players could have become more engaged [55] and wanted to keep playing. Also, with Modeling, players were already taught how to score well, and that could have motivated players to try their best, as they already knew what to do. For the players in the Instructing condition, these motivations were missing, and therefore it was very likely that players became less motivated as they did not know how to score well, and they were provided minimal feedback on whether or not they were doing the right thing. This lack of guidance and feedback may have been very demoralizing for the players, ultimately leading

to lower Adherence.

Although this study used a game to solicit specific player behaviors, the findings can be generalized to inform the design of general gesture-based interfaces for improving Learnability and Adherence. Pertaining to Learnability, the lack of visibility of physical affordances can be mitigated through Modeling, as providing a model for imitation primes the user to expect a specific mental model of how the system works. This model should be as complete as possible, requiring users to “fill in the blanks” as little as possible, especially for complex interactions and concepts. Providing timely feedback can also help users build this mental model through trial and error, but this method of scaffolding is less efficient for more complex interactions than Modeling. In terms of Adherence, managing user expectation is very important, and providing users with Modeling and Feeding-back features is pertinent for keeping users motivated and sustaining interest in the task.

6.5 Conclusions and Future Work

In this study, I endeavored to collect a large amount of gesture data in different scaffolding conditions (Modeling, Feeding-back, Instructing) to determine which scaffolding condition was most suitable for improving the Learnability of a mid-air gesture-based system and enhancing passers-by’ Adherence to the system facilitated task in a public-situated display. To achieve this, I built a gesture-based game using the Leap Motion Controller as input, decomposing this input down into Atomic Gestures, and storing the Composite representation of these gestures into a relational database. These representations then formed the foundation of a gesture recognition system. I then utilized this system to design and deploy an exhibit in a public space for the purposes of data collection in an experiment. The results of the experiment showed that Modeling was most effective and efficient at improving Learnability and Adherence, followed by Feeding-back, while Instructing had no effect. I then discussed the possible reasons for this, ultimately attributing the cause to mental-model building for Learnability, and providing motivation for Adherence.

The study focused on only three types of Scaffolding Means. Future work should expand the focus to other scaffolding conditions. The study used a very specific method of parameterizing and measuring the concepts. Perhaps other methods of measurement may be found in future studies. Also, the study described a very specific type of exhibit and game design, and whether or not

the proposed methods of investigation will work for other designs and layouts remains a potential avenue for future inquiry.



Conclusion

7.1 Summary

The increasing demand in consumer products has resulted number of display technologies. Presence of a third dimension have added a feeling of reality to the two dimensional images. Therefore researcher have been developing 3D displays to provide more realistic imagery for more than a half century. These 3D displays rely on one of the most amazing properties of the human vision which is ability to perceive the depth of images being viewed. There are variates of 3D displays from stereoscopic displays using special glasses, head mounded displays, or glasses-free displays. Among all types of 3D displays including volumetric, large 3D stereoscopic and personal head-mounted displays (HMDs); the stereoscopic displays and HMDs are most common types of 3D displays due to their applications in PCs and smartphones. Unlike to typical 3D displays, which we use in home, the interactive displays provide interaction between the users and content screened on the display by tracking the position of user's fingers/hand, and head.

Having an interactive 3D display allows users to intuitively interact with virtual objects freely move in a mid-air by using 3D tracking technologies track a user's hand or finger motions. 3D interactive displays have a large variety of forms and support a wide range of 3D mid-air interaction techniques and can be used for single and multi-user interactions. Moreover, public-exhibited displays, which address also multi-user to observe interaction, particularly large 2D displays, also have been increasingly deployed in various (semi-) public spaces, including galleries, schools and shopping centers. While touch interaction is common modality for these displays, mid-air interaction with such

displays has several advantages such as being hygienic, allowing to interact with large-size displays, and enabling interaction without come close to display. Therefore, I focused on investigation of 3D mid-air interaction for these three widespread single and multi-user display settings: personal, collaborative, and public-exhibited displays. Since each display setting suffers from different issues regarding 3D interaction, I addressed the issues and aimed to offer the solutions for each setting.

In Chapter 1, I provided properties of 3D displays and variates forms of them. As above mentioned, these displays can be used by a single user for personal use and/or multiple users for collaborative tasks. I discussed the interaction way and proposed several research question regarding 3D interaction with them which have not been answered in association with these 3D displays.

In Chapter 2, I provided a review of background literature. I discussed 3D display technologies, devices for 3D input, 3D mid-air interaction techniques, 3D user interfaces, and collaborative interactions which help us to better understand building new 3D interaction techniques and user interfaces.

In Chapter 3 and 4, I focused on 3D selection techniques for personal HMDs. I explored how different selection techniques can be integrated into high-level user interfaces in personal HMDs with low-cost built-in tracking sensors.

In Chapter 3, I presented a raycasting technique for 3D interactions in self-contained Augmented Reality (AR) head-mounted displays. Unlike traditional raycasting which requires absolute spatial and rotational tracking of a user's limb, my approach relies on the relative rotation values captured by an inertial measurement unit (IMU) on a hand-held controller (wand). However, these values cannot be directly mapped to the ray direction. Therefore, I addressed a number of challenges to enable IMU-based raycasting, and reported the results of two user studies. These studies contribute adoption of raycasting-based interaction for stereoscopic HMDs.

In Chapter 4, I explored 3D interaction techniques in HMDs for mobile selection tasks. Explored selection techniques solely rely on tracking technologies built into conventional commercially-available HMDs (i.e. gesture trackers, gaze tracking, and inertial measurement units). While two techniques are based on raycasting using built-in inertial measurement unit (IMU) sensing, three techniques are based on a hand-controlled 3D cursor using gestural tracking. I conducted a user study to evaluate the techniques for 3D object selection within arm's reach, and discussed implications of findings for design of selection techniques in mobile HMDs.

In Chapter 5, I designed and implemented a novel, thin, and compact interactive multi-user stereoscopic tabletop display using a 3D TV for a typical collaborative workspace. Moreover, I also implemented a finger-based direct interface for object manipulation in a multi-user display system, in which they can share the manipulations from their own point of view. I conducted an empirical comparison of my direct interaction interface to an indirect remote pointing method. Also, I have utilized proposed interactive display to create bi-directional collaborative interaction workspaces for remote modeling tasks in which a physical interactive space enables a user to directly modify a fully tracked custom-shaped object, and another remote interactive space provides a direct 3D interface enabling users to view and interact with this object in real-time through its rendered virtual shape in 3D display.

In Chapter 6, I focused on important challenge is that lack of design insight on how passers-by start interactions and how they use simple mid-air gestures in such displays. I aimed to address how to design 3D mid-air interface for passers-by around public-exhibited displays by designing an interactive game, which is played with simple mid-air gestures. I collected mid-air gesture data to find how passers-by interact with situated displays by analyzing effect of different types of instructions on learning performance of passers-by.

7.2 Research Contributions

This dissertation covers a range of works to interaction techniques and to design 3D user interfaces for mid-air interaction for single and multi-user displays. A summary of the contributions is give below.

7.2.1 Contributions

- A novel raycasting technique called GyroWand which enables raycasting in stereoscopic HMDs using IMU rotational data from a handheld controller (Section 3.3.2).
- The development of three novel disambiguation methods which address the ambiguity problem associated with raycasting in HMDs using IMU rotational data from a handheld controller (Section 3.5).
- Presenting results of two controlled user studies revealing the most suitable design parameters: the origin of the virtual ray (Section 3.4.1) and the disambiguation mechanism (Section 3.5.4).

- An exploration of 3D selection techniques rely solely on built-in sensors in Optical HMDs for mobile usage scenarios. A user study evaluating five techniques for 3D object selection within arm's reach and discuss implications of findings for design of selection techniques in mobile HMDs. (Chapter 4).
- Designing and implementing a novel flexible interactive multi-user stereoscopic display using a 3D TV for collaborative manipulation tasks (Section 5.3).
- Designing and implementing direct mid-air interaction techniques to enable users to directly select and manipulate the 3D objects through this display (Section 5.5).
- Creating a bi-directional remote collaborative workspaces in which a robot arm and projection mapping in the physical interaction space reflect the manipulations performed remotely in the virtual interaction space (Section 5.7).
- Using embedded magnetic sensors and a sensing table to extract 3D shape of a custom-shape physical object in real time while one is directly manipulating it by fingers (Section 5.7.2).
- Building a mid-air gesture-based game and decomposing these gestures down into Atomic Gestures, and the Composite representation of them into a relational database.
- Providing a set of guidance for passers-by to learn and engage interaction with a public-exhibited display using 3D mid-air gestures (Section 6.2).

7.2.2 Future Directions

In this dissertation, I aimed to investigate interaction issues, in an attempt to lay the groundwork for future research and application development using mid-air interaction with single and multi-user displays. This work provides some important initial steps towards making these displays interactive 3D platforms. Through this research I have identified a number of future areas of research.

Direct mid-air interaction may cause arm fatigue

Direct mid-air interaction techniques using hand gestures I used in my studies

required large hand and arm gestures performed in front of body. The problem is that large motions were tiring because it required more physical movement to perform interaction. However, mid-air interaction techniques with hand-held device I evaluated required relatively smaller hand gestures, therefore, more comfortable. Considering interaction in mobile scenarios, using large hand gestures could be also socially conspicuous. Therefore instead of using mid-air hand gestures, constrained hand movements, such as arms-down mid-air techniques may deliver low arm fatigue and social acceptance issues for hand-controlled 3D cursor techniques [54].

Mid-air interaction without tactile feedback my feel unreal

3D displays show 3D virtual object in air and enable users to see them from different points of view. Hand gestures are natural and intuitive way to select 3D objects in air. In order to confirm the selection, visual feedback could be used, however, user's finger may occlude the object in front of the finger. Therefore, I believe that tactile feedback is an important feedback mechanism when interacting 3D virtual objects. Tactile feedback can make the interaction more realistic and gives feeling that hands contacted with the virtual objects [145].

Confliction in simultaneous mid-air interactions

Direct mid-air interface in collaborative 3D displays enable multiple users to interact with virtual contents. However, using a common interaction space for input collaborators causes the concurrency control problems [43]. The following solutions could be used to avoid conflicting interaction such as using locking mechanisms, following social protocols [43], or coordination policies [113] to enable collaborative mid-air interactions.

Begin-end mid-air interaction with displays in public spaces

Techniques that require a dedicated device can have virtual or physical button to trigger selection. However, for mid-air interaction, several gestures have been proposed to trigger a selection: the most often used is the pinch gesture, pinching index thumb fingers to trigger an action [22, 34, 42]; other techniques such as AirTap [29], ThumbTrigger [46], and SideTrigger [11] use different gestures. Therefore, which gesture is the best in terms of easiness and intuitiveness, particularly for public-exhibited displays, should be examined.

7.3 Conclusion

Nowadays, a wide range of fields such as entertainment, visualization, education, industrial design has increasingly required interactive displays technologies. These displays vary from 2D to 3D, as well as according to intended use such as for personal, collaborative or exhibited displays in public space. Thanks to various types motion tracking systems with various tracking approaches, which support various ranges, such as a rooms-size tracking using situated depth cameras, and finger/hand tracking using a mobile tracker, such displays have been more interactive in any forms, and enabling users to interact with content in three-dimensional space. Although these technologies provide a wide range of design possibilities, designing relevant 3D mid-air interfaces is still challenging due to substantial dependencies regarding displays and tracking systems.

In this dissertation, I focused on 3D mid-air interaction issues associated with different types single and multi-user displays: personal, collaborative, and public-exhibited displays. Firstly, I address issues with see-through head mounted display, one of the most popular personal 3D displays particularly used in mobile scenarios. Therefore, I first explore and investigate fundamental performances of commonly used and promising selection techniques designed for HMDs with limited sensing capabilities of built-in trackers.

My second work is regarding collaborative 3D displays, I newly designed a thin and compact, and flexible IllusionHole using a 3D TV. Also, the directness offered by IllusionHole led to create direct finger-based 3D interfaces, by which multiple viewers can directly point out, select and manipulate a shared 3D content from their individual point of view around the multi-user 3D tabletop display. Result of a user study showed that proposed finger-based direct interface provides faster content selection and manipulation than 3D cursor technique. I then utilized this direct interface for remote collaboration through IllusionHole for physical object modelling task. The proposed direct interface successfully enabled a group of users around IllusionHole to manipulate the virtual form of physical models to remotely contribute to the modelling task.

Second, I focused on 3D mid-air interaction issues collaborative multi-user 3D displays that enable a group of users to share and manipulate displayed content through various modalities for collaborative design, entertainment, data explorations while having face-to-face formation. I newly designed a thin and compact, and flexible collaborative display by using a 3D TV and aimed

to provide deep insights of interaction techniques how to directly select and manipulate virtual object among multiple users. Third, I focused on 3D mid-air interaction issues public-exhibited large display in various public spaces, including galleries, schools and shopping centers and so on. 3D mid-air interaction with such displays is still challenging due to lack of insight into design of mid-air interactions for passers-by around the public display.

I have presented an overall exploration of interaction issues associated with single and multi-user displays. I first designed new 3D interaction techniques and user interfaces based on experimental data. I also designed and implemented multi-user interactive stereoscopic displays for collaborative scenarios and designed suitable interaction techniques for them. I also have discussed the lessons learned for the adoption of interaction techniques in such displays. This work can thus serve as a baseline for future experiments, 3D interaction techniques, and user interfaces to build on personal, collaborative, and public-exhibited displays.

Bibliography

- [1] Kasim Ozacar, Kazuki Takashima, and Yoshifumi Kitamura. “Direct 3D Object Manipulation on a Collaborative Stereoscopic Display”. In: *Proceedings of ACM Symposium on Spatial User Interaction*. 2013, pp. 69–72.
- [2] Adi Abileah. “Overview: Testing 3D-stereo Displays, Techniques and Challenges”. In: *SID Symposium Digest of Technical Papers* 44.1 (2013), pp. 368–371.
- [3] Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. “The Two-user Responsive Workbench: Support for Collaboration Through Individual Views of a Shared Space”. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM. 1997, pp. 327–332.
- [4] Laurent Aguerreche, Thierry Duval, Anatole Lécuyer, and Bruno Arnaldi. “Demonstration: 3-Hand Manipulation of Virtual Objects”. In: *Proceedings of the Joint Virtual Reality Conference of EGVE - ICAT - EuroVR*. Eurographics. 2005, 2 pages.
- [5] Kurt Akeley, Simon J. Watt, Ahna Reza Girshick, and Martin S. Banks. “A Stereo Display Prototype with Multiple Focal Distances”. In: *ACM Trans. Graph.* 23.3 (2004), pp. 804–813.
- [6] David Anderson, James L. Frankel, Joe Marks, Aseem Agarwala, Paul Beardsley, Jessica Hodgins, Darren Leigh, Kathy Ryall, Eddie Sullivan, and Jonathan S. Yedidia. “Tangible Interaction + Graphical Interpreta-

- tion: A New Approach to 3D Modeling”. In: *Proceedings of the Conference on Computer Graphics and Interactive Techniques*. 2000, pp. 393–402.
- [7] Gillian Andrews. “Dance Dance Revolution”. In: *Space Time Play* (2007), pp. 20–21.
 - [8] Ferran Argelaguet Sanz and Carlos Andujar. “A Survey of 3D Object Selection Techniques for Virtual Environments”. In: *Computers and Graphics* 37.3 (2013), pp. 121–136.
 - [9] Emilio Camahort Armando de la Re Eduardo Martorell and Francisco Abad. “Inexpensive 3D Stereo for Two Users Using a Single Display”. In: *Proceedings of Ibero-American Symposium in Computer Graphics*. 2011.
 - [10] Elliott M Avedon and Brian Sutton-Smith. *The Study of Games*. Wiley, New York, 1971.
 - [11] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. “Pointable: An In-air Pointing Technique to Manipulate Out-of-reach Targets on Tabletops”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM. 2011, pp. 11–20.
 - [12] Ari Y. Benbasat and Joseph A. Paradiso. “An Inertial Measurement Framework for Gesture Recognition and Applications”. In: *Proceedings of International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*. Springer-Verlag. 2002, pp. 9–20.
 - [13] H. Benko and S. Feiner. “Balloon Selection: A Multi-Finger Technique for Accurate Low-Fatigue 3D Selection”. In: *Proceedings of the IEEE Symposium on 3D User Interfaces*. 2007.
 - [14] Maurice H. P. H. van Beurden, Wijnand A. Ijsselstein, and Yvonne A. W. de Kort. “User Experience of Gesture Based Interfaces: A Comparison with Traditional Interaction Methods on Pragmatic and Hedonic Qualities”. In: *Gesture and Sign Language in Human-Computer Interaction and Embodied Communication*. Vol. 7206. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 36–47.
 - [15] Eric A Bier and Steven Freeman. “MMM: A User Interface Architecture for Shared Editors On a Single Screen”. In: *Proceedings of the 4th ACM Symposium on User Interface Software and Technology*. 1991, pp. 79–86.

- [16] John Bolton, Kibum Kim, and Roel Vertegaal. "SnowGlobe: A Spherical Fish-Tank VR Display". In: *Proceedings of the Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2011, pp. 1159–1164.
- [17] Benoit Bossavit, Jean-Baptiste de la Rivière, Toni Da Luz, Mathieu Courtois, Cédric Kervegant, and Martin Hachet. "An Immersive Multitouch Workspace". In: *Proceedings of ACM SIGGRAPH Emerging Technologies*. ACM. 2011, 1 page.
- [18] R Bourdon. "WampServer, a Windows web development environment". In: *Retrieved June 25 (2013)*, p. 2013.
- [19] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. "Testbed Evaluation of Virtual Environment Interaction Techniques". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM, 1999, pp. 26–33.
- [20] Doug A Bowman, Donald B Johnson, and Larry F Hodges. "Testbed Evaluation of Virtual Environment Interaction Techniques". In: *Presence* 10.1 (2001), pp. 75–95.
- [21] Doug A Bowman, Ernst Kruijff, Joseph J LaViola Jr, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, 2004.
- [22] Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Ly. "Using Pinch Gloves for Both Natural and Abstract Interaction Techniques in Virtual Environments". In: *Proceedings of HCI International*. 2001, pp. 629–633.
- [23] O. Cakmakci and Jannick Rolland. "Head-worn Displays: A Review". In: *Journal of Display Technology* 2.3 (Sept. 2006), pp. 199–216.
- [24] Yeonjoo Cha and Rohae Myung. "Extended Fitts' Law for 3D Pointing Tasks using 3D Target Arrangements". In: *International Journal of Industrial Ergonomics* 43.4 (2013), pp. 350–355.
- [25] Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. "Touching the Void: Direct-touch Interaction for Intangible Displays". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 2625–2634.

- [26] Shrikant Chaudhary and Prasad Rane. *World 3D Display Market - Opportunities and Forecasts, 2013 - 2020*. 2015. URL: <http://www.alliedmarketresearch.com/3d-display-market> (visited on 07/14/2016).
- [27] Mingyu Chen, G. AlRegib, and Biing-Hwang Juang. "A New 6D Motion Gesture Database and the Benchmark Results of Feature-based Statistical Recognition". In: *Proceeding of IEEE International Conference on Emerging Signal Processing Applications*. 2012, pp. 131–134.
- [28] Dewen Cheng, Yongtian Wang, Hong Hua, and M. M. Talha. "Design of an Optical See-through Head-mounted Display with a Low f-number and Large Field of View using a Freeform Prism". In: *Appl. Opt.* 48.14 (May 2009), pp. 2655–2668.
- [29] A. Cockburn, P. Quinn, C. Gutwin, G. Ramos, and J. Looser. "Air Pointing: Design and Evaluation of Spatial Target Acquisition with and Without Visual Feedback". In: *Int. J. Hum.-Comput. Stud.* 69.6 (June 2011), pp. 401–414.
- [30] Dane Coffey, Nicholas Malbraaten, Trung Le, Iman Borazjani, Fotis Sotiropoulos, and Daniel F. Keefe. "Slice WIM: A Multi-Surface, Multi-Touch Interface for Overview+Detail Exploration of Volume Datasets in Virtual Reality". In: *Proceedings of Symposium on Interactive 3D Graphics and Games*. ACM. 2011, pp. 191–198.
- [31] Stephen DiVerdi, Aravind Krishnaswamy, and Sunil Hadap. "Industrial-Strength Painting with a Virtual Bristle Brush". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 2010, pp. 119–126.
- [32] Johan Partomo Djajadiningrat. *Cubby: What You See Is Where You Act. Interlacing the Display and Manipulation Spaces*. TU Delft, Delft University of Technology, 1998.
- [33] David Ebert, Edward Bedwell, Stephen Maher, Laura Smoliar, and Elizabeth Downing. "Realizing 3D Visualization Using Crossed-Beam Volumetric Displays". In: *Commun. ACM* 42.8 (1999).
- [34] E.A. Edirisinghe, J. Jiang, and E. A. Edirisinghej. Jiang. *Stereo Imaging, An Emerging Technology*. 2000.

- [35] Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. “Ethereal Planes: A Design Framework for 2D Information Spaces in 3D Mixed Reality Environments”. In: *Proceedings of the ACM symposium on Spatial user interaction*. 2014, pp. 2–12.
- [36] Epson Moverio. 2015. URL: <http://www.epson.com/moverio>.
- [37] Vasily A. Ezhov. “Invited Paper:3D Displays:Fundamental Physical Classification for Clarifying Inherent Technical Features”. In: *EuroDisplay 2007* (2007).
- [38] Paul M Fitts. “The information Capacity of the Human Motor System in Controlling the Amplitude of Movement.” In: *Journal of experimental psychology* 47.6 (1954), pp. 381–391.
- [39] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. “Aperture Based Selection for Immersive Virtual Environments”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM. 1996, pp. 95–96.
- [40] Chi-Wing Fu, Wooi-Boon Goh, and Junxiang Allen Ng. “Multi-Touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations”. In: *Proceedings of the International Conference on Human Factors in Computing Systems*. ACM. 2010, pp. 2213–2222.
- [41] Pragati Garg., Naveen Aggarwal., and Sanjeev Sofat. “Vision Based Hand Gesture Recognition”. In: *World Academy of Science, Engineering and Technology*. 2009.
- [42] Jason Geng. “Three-dimensional Display Technologies”. In: *Adv. Opt. Photon.* 5.4 (Dec. 2013), pp. 456–535.
- [43] Saul Greenberg and David Marwood. “Real-time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface”. In: *Proceedings of the ACM conference on Computer supported cooperative work*. 1994, pp. 207–217.
- [44] T. Grossman and D. Wigdor. “Going Deeper: a Taxonomy of 3D on the Tabletop”. In: *Proceedings on IEEE International Workshop on Horizontal Interactive Human-Computer Systems*. 2007, pp. 137–144.

- [45] Tovi Grossman and Ravin Balakrishnan. “Collaborative Interaction with Volumetric Displays”. In: *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 383–392.
- [46] Tovi Grossman and Ravin Balakrishnan. “The Design and Evaluation of Selection Techniques for 3D Volumetric Displays”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM. 2006, pp. 3–12.
- [47] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. “Multi-Finger Gestural Interaction with 3D Volumetric Displays”. In: *ACM Trans. Graph.* 24.3 (2005), pp. 931–931.
- [48] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. “Multi-finger Gestural Interaction with 3D Volumetric Displays”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM. 2004, pp. 61–70.
- [49] Gerwin De Haan, Michal Koutek, and Frits H. Post. “IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection”. In: *Proceedings of Virtual Environments*. 2005, pp. 201–209.
- [50] Martin Hachet, Benoit Bossavit, Aurélie Cohé, and Jean-Baptiste de la Rivière. “Toucheo: Multitouch and Stereo Combined in a Seamless Workspace”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 2011, pp. 587–592.
- [51] G. Hackenberg, R. McCall, and W. Broll. “Lightweight Palm and Finger Tracking for Real-time 3D Gesture Control”. In: *Proceedings of IEEE Virtual Reality Conference*. 2011, pp. 19–26.
- [52] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. “Shallow-Depth 3D Interaction: Design and Evaluation of One-, Two- and Three-Touch Techniques”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2007, pp. 1147–1156.
- [53] Mark Hancock, Thomas ten Cate, and Sheelagh Carpendale. “Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. 2009, pp. 133–140.

- [54] Faizan Haque, Mathieu Nancel, and Daniel Vogel. "Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors". In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 3653–3656.
- [55] John Hattie and Helen Timperley. "The Power of Feedback". In: *Review of Educational Research* 77.1 (2007), pp. 81–112.
- [56] Pierre St. Hilaire, Stephen A. Benton, Mark Lucente, Mary Lou Jepsen, Joel Kollin, Hiroshi Yoshikawa, and John Underkoffler. "Electronic Display System for Computational Holography". In: *Proceedings of the SPIE on Practical Holography*. 1990, pp. 1212–1220.
- [57] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. "Interactions in the Air: Adding Further Depth to Interactive Tabletops". In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM, 2009, pp. 139–148.
- [58] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. "HoloDesk: Direct 3D Interactions with a Situated See-Through Display". In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 2012, pp. 2421–2430.
- [59] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. "Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 1063–1072.
- [60] Juan David Hincapié-Ramos, Kasim Ozacar, Pourang P. Irani, and Yoshifumi Kitamura. "GyroWand: IMU-based Raycasting for Augmented Reality Head-Mounted Displays". In: *Proceedings of the ACM Symposium on Spatial User Interaction*. 2015, pp. 89–98.
- [61] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. "Passive Real-World Interface Props for Neurosurgical Visualization". In: *Proceedings of Conference Companion on Human Factors in Computing Systems*. ACM. 1994, pp. 232–.
- [62] David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. "Vergence-accommodation Conflicts Hinder Visual Performance and Cause Visual Fatigue". In: *Journal of Vision* 8 (3 2008), pp. 33–33.

- [63] M. Hoffman, P. Varcholik, and J.J. LaViola. "Breaking the Status Quo: Improving 3D gesture Recognition with Spatially Convenient Input Devices". In: *Proceeding of IEEE Virtual Reality Conference*. 2010, pp. 59–66.
- [64] Errol R Hoffmann, Colin G Drury, and Carol J Romanowski. "Performance in One-, Two-and Three-dimensional Terminal Aiming Tasks". In: *Ergonomics* 54.12 (2011), pp. 1175–1185.
- [65] Pengyu Hong, M. Turk, and T.S. Huang. "Gesture Modeling and Recognition using Finite State Machines". In: *Proceeding of 4th IEEE International Conference on Automatic Face and Gesture Recognition*. 2000, pp. 410–415.
- [66] Jiawei Huang, Kazuki Takashima, Shuichiro Hashi, and Yoshifumi Kitamura. "IM3D: Magnetic Motion Tracking System for Dexterous 3D Interactions". In: *ACM SIGGRAPH 2014 Emerging Technologies*. 2014, 1 page.
- [67] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. "Direct Manipulation Interfaces". In: *International Journal of Human-Computer Interaction* 1.4 (1985), pp. 311–338.
- [68] H. Ichida, Y. Itoh, Y. Kitamura, and F. Kishino. "Interactive Retrieval of 3D virtual Shapes using Physical Objects". In: *Proceeding of IEEE Virtual Reality*. 2004, pp. 231–232.
- [69] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon. "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 2011, pp. 559–568.
- [70] Robert J. K. Jacob. "The Use of Eye Movements in Human-computer Interaction Techniques: What You Look at is What You Get". In: *ACM Trans. Inf. Syst.* 9.2 (Apr. 1991), pp. 152–169.
- [71] McGonigal Jane. "Reality is Broken: Why Games Make Us Better and How They Can Change the World". In: *Reality is Broken: Why Games Makes Us Better and How They Can Change the World* (2011), pp. 22–23.

- [72] Xiaodan Jiao, Hui Deng, and Feng Wang. "An Investigation of Two-Handed Manipulation and Related Techniques in Multi-Touch Interaction". In: *Proceedings of the International Conference on Machine Vision and Human-Machine Interface*. 2010, pp. 565–568.
- [73] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. "A Comparison of Ray Pointing Techniques for Very Large Displays". In: *Proceedings of Graphics Interface*. Canadian Information Processing Society. 2010, pp. 269–276.
- [74] Hideki Kakeya, Shuta Ishizuka, and Yuya Sato. "Realization of an Aerial 3D Image that Occludes the Background Scenery". In: *Opt. Express* 22.20 (Oct. 2014), pp. 24491–24496.
- [75] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. "Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor". In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 2012, pp. 167–176.
- [76] Hidei Kimura, Taro Uchiyama, and Hiroyuki Yoshikawa. "Laser Produced 3D Display in the Air". In: *Proceedings of ACM SIGGRAPH Emerging Technologies*. 2006.
- [77] Yoshifumi Kitamura, Tetsuya Kikukawa, Satoshi Mieda, Yotaka Kunita, Megumi Isogai, and Hideaki Kimata. "Image-based 3D Telecopier: A System for Sharing a 3D Object by Multiple Groups of People at Remote Locations". In: *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*. 2010, pp. 51–54.
- [78] Yoshifumi Kitamura, Takashige Konishi, Sumihiko Yamamoto, and Fumio Kishino. "Interactive Stereoscopic Display for Three or More Users". In: *Proceedings of the Conference on Computer Graphics and Interactive Techniques*. ACM. 2001, pp. 231–240.
- [79] Yoshifumi Kitamura, Tomokazu Nakayama, Takashi Nakashima, and Sumihiko Yamamoto. "The Illusionhole with Polarization Filters". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 2006, pp. 244–251.

- [80] Sebastian Knoedel and Martin Hachet. "Multi-touch RST in 2D and 3D Spaces: Studying the Impact of Directness on User Performance". In: *Proceedings of IEEE Symposium on 3D User Interfaces*. 2011, pp. 75–78.
- [81] Kishore Reddy Konda, Achim Königs, Hannes Schulz, and Dirk Schulz. "Real Time Interaction with Mobile Robots Using Hand Gestures". In: *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*. 2012, pp. 177–178.
- [82] Alexander Kulik, André Kunert, Stephan Beck, Roman Reichel, Roland Blach, Armin Zink, and Bernd Froehlich. "C1x6: A Stereoscopic Six-user Display for Co-located Collaboration in Shared Virtual Environments". In: *Proceedings of the SIGGRAPH Asia Conference*. ACM. 2011, pp. 188–200.
- [83] Joseph J. LaViola and Daniel F. Keefe. "3D Spatial Interaction: Applications for Art, Design, and Science". In: *Proceedings of ACM SIGGRAPH Courses*. 2011, pp. 1–75.
- [84] JaeYeol Lee, GueWon Rhee, and DongWoo Seo. "Hand Gesture-based Tangible Interactions for Manipulating Virtual Objects in a Mixed Reality Environment". In: *The International Journal of Advanced Manufacturing Technology* 51.9-12 (2010), pp. 1069–1082.
- [85] Min Kyung Lee and Leila Takayama. "'Now, I Have a Body': Uses and Social Norms for Mobile Remote Presence in the Workplace". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 33–42.
- [86] SangYoon Lee, Jinseok Seo, Gerard J Kim, and Chan-Mo Park. "Evaluation of Pointing Techniques for Ray Casting Selection in Virtual Environments". In: *Proceedings of International Conference on Virtual Reality and its Application in Industry*. 2003, pp. 38–44.
- [87] Taehee Lee and Tobias Hollerer. "Handy AR: Markerless Inspection of Augmented Reality Objects using Fingertip Tracking". In: *Proceedings on International Symposium on Wearable Computers*. IEEE. 2007, pp. 83–90.

- [88] B. Leibe, T. Starner, W. Ribarsky, Z. Wartell, D. Krum, B. Singletary, and L. Hodges. "The Perceptive Workbench: Toward Spontaneous and Natural Interaction in Semi-immersive Virtual Environments". In: *Proceedings of IEEE Virtual Reality*. 2000, pp. 13–20.
- [89] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. "Physical Telepresence: Shape Capture and Display for Embodied, Computer-mediated Remote Collaboration". In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. 2014, pp. 461–470.
- [90] Dmitri K Lemmerman and Joseph J LaViola Jr. "Effects of Interaction-Display Offset on User Performance in Surround Screen Virtual Environments". In: *Proceedings of IEEE Virtual Reality*. 2007, pp. 303–304.
- [91] Zhi Li and Ray Jarvis. "Real-time Hand Gesture Recognition using a Range Camera". In: *Proceedings of Australasian Conference on Robotics and Automation*. 2009, pp. 529–534.
- [92] Jiandong Liang and Mark Green. "JDCAD: A Highly Interactive 3D Modeling System". In: *Computers & Graphics* 18.4 (1994), pp. 499–506.
- [93] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. "Gunslinger: Subtle Arms-down Mid-air Interaction". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 63–71.
- [94] R Lopez-Gulliver, S Yoshida, S Yano, and N Inoue. "gCubik: A Cubic Autostereoscopic Display for Multiuser Interaction: Grasp and Group-Share Virtual Images". In: *Proceedings of International Conference on Computer Graphics and Interactive Techniques*. ACM, 2008, pp. 5–5.
- [95] Weiquan Lu, Mandi Jieying Lee, Teong Leong Chuah, Chun Kit Lee, Zheng Yi Lim, and Ellen Yi-Luen Do. "WildAR: Creating a Networked AR System for "In-the-Wild" Studies". In: *Proceedings of International Symposium on Mixed and Augmented Reality-Media, Art, Social Science, Humanities and Design*. IEEE. 2015, pp. 34–39.
- [96] I Scott MacKenzie. "Fitts' Law as a Research and Design Tool in Human-Computer Interaction". In: *Human-Computer Interaction* 7.1 (1992), pp. 91–139.
- [97] I. Scott MacKenzie and William Buxton. "Extending Fitts' Law to Two-dimensional Tasks". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1992, pp. 219–226.

- [98] S.O.H. Madgwick, A.J.L. Harrison, and R. Vaidyanathan. “Estimation of IMU and MARG orientation using a gradient descent algorithm”. In: *Proceedings of IEEE International Conference on Rehabilitation Robotics*. June 2011, pp. 1–7.
- [99] Shahzad Malik and Joe Laszlo. “Visual Touchpad: A Two-Handed Gestural Input Device”. In: *Proceedings of the International Conference on Multimodal Interfaces*. ACM. 2004, pp. 289–296.
- [100] Jonathan Marbach. “GPU Acceleration of Stereoscopic and Multi-View Rendering for Virtual Reality Applications”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 2009, pp. 103–110.
- [101] Paul Marshall, Richard Morris, Yvonne Rogers, Stefan Kreitmayer, and Matt Davies. “Rethinking ‘Multi-user’: An In-the-wild Study of How Groups Approach a Walk-up-and-use Tabletop Interface”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 3033–3042.
- [102] Anthony Martinet, Géry Casiez, and Laurent Grisoni. “The Effect of DOF Separation in 3D manipulation Tasks with Multi-touch Displays”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 2010, pp. 111–118.
- [103] D. Mendes, F. Fonseca, B. Araùjo, A. Ferreira, and J. Jorge. “Mid-air Interactions Above Stereoscopic Interactive Tables”. In: *Proceeding of the IEEE Symposium on 3D User Interfaces*. 2014, pp. 3–10.
- [104] *META SpaceGlassess*. 2015. URL: <https://www.getameta.com/>.
- [105] *Metaio SDK*. 2015. URL: <http://www.metaio.com/products/sdk/>.
- [106] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. “Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements.” In: *Psychological Review* 95.3 (1988), pp. 340–370.
- [107] Microsoft. *Kinect for xbox360*. <http://www.xbox.com/en-US/kinect>. 2010.
- [108] *Microsoft HoloLens*. 2015. URL: <https://www.microsoft.com/microsofthololens/en-us/>.

- [109] Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, and Joseph J. LaViola Jr. "Interactive 3D Model Acquisition and Tracking of Building Block Structures." In: *IEEE Trans. Vis. Comput. Graph.* 18.4 (2012), pp. 651–659.
- [110] Mark R. Mine. *Virtual Environment Interaction Techniques*. Tech. rep. University of North Carolina at Chapel Hill, 1995, 18 pages.
- [111] Mark R. Mine, Frederick P. Brooks Jr., and Carlo H. Sequin. "Moving Objects in Space: Exploiting Proprioception in Virtual-environment Interaction". In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. 1997, pp. 19–26.
- [112] Pranav Mistry and Pattie Maes. "SixthSense: A Wearable Gestural Interface". In: *Proceedings of ACM SIGGRAPH ASIA Sketches*. 2009, 1 page.
- [113] Meredith Ringel Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. "Beyond Social Protocols: Multi-user Coordination Policies for Co-located Groupware". In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 2004, pp. 262–265.
- [114] Tomer Moscovich and John F. Hughes. "Indirect Mappings of Multi-Touch Input Using One and Two Hands". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2008, pp. 1275–1284.
- [115] Christian Müller-Tomfelde. "Dwell-based Pointing in Applications of Human Computer Interaction". In: *Proceedings of the INTERACT*. Springer-Verlag. 2007, pp. 560–573.
- [116] Alessandro Mulloni, Hartmut Seichter, and Dieter Schmalstieg. "Hand-held Augmented Reality Indoor Navigation with Activity-based Instructions". In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*. ACM, 2011, pp. 211–220.
- [117] Brad A Myers, Herb Stiel, and Robert Gargiulo. "Collaboration Using Multiple PDAs Connected to a PC". In: *Proceedings of the ACM Conference on Computer Supported Cooperative Work*. 1998, pp. 285–294.

- [118] Kousuke Nakashima, Takashi Machida, Kiyoshi Kiyokawa, and Haruo Takemura. "A 2D-3D Integrated Environment for Cooperative Work". In: *Proceedings of the ACM Symposium on Virtual reality Software and Technology*. ACM. 2005, pp. 16–22.
- [119] Y. Nakazato, M. Kanbara, and N. Yokoya. "Wearable Augmented Reality System using Invisible Visual Markers and an IR Camera". In: *Proceedings of 9th IEEE International Symposium on Wearable Computers*. 2005, pp. 198–199.
- [120] Tek-Jin Nam and David Wright. "The Development and Evaluation of Syco3D: a Real-time Collaborative 3D CAD System". In: *Design Studies* 22.6 (2001), pp. 557–582.
- [121] Alexis Paljic Nan Wang and Philippe Fuchs. "A Study of Perception of Volumetric Rendering for Immersive Scientific Visualization". In: *Proceedings of International Conference on Artificial Reality and Telexistence*. 2010, pp. 145–152.
- [122] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. "Mid-Air Pan-and-Zoom On Wall-Sized Displays". In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 2011, pp. 177–186.
- [123] NASA-TLX: *Task Load Index Desktop Application Instructions (v.0.5)*. <http://www.playgraph.com/apps/nasa-tlx>. 2012.
- [124] Donald A Norman. "Natural User Interfaces Are Not Natural". In: *Interactions* 17.3 (2010), pp. 6–10.
- [125] Alex Olwal, Hrvoje Benko, and Steven Feiner. "SenseShapes: Using Statistical Geometry for Object Selection in a Multimodal Augmented Reality System". In: *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*. 2003, pp. 300–301.
- [126] Alex Olwal and Steven Feiner. "The Flexible Pointer: An Interaction Technique for Augmented and Virtual Reality". In: *Proceedings of ACM Symposium on User Interface Software and Technology*. 2003, pp. 81–82.
- [127] Michael Ortega and Laurence Nigay. "AirMouse: Finger Gesture for 2D and 3D Interaction". In: *Proceedings of Conference on Human-Computer Interaction*. 2009, pp. 214–227.

- [128] Alex Pang and Craig Wittenbrink. "Collaborative 3D visualization with CSpray". In: *IEEE Computer Graphics and Applications* 17.2 (1997), pp. 32–41.
- [129] R. E. Patterson. "Interactive Stereoscopic Displays". In: *Journal of the Society for Information Display* (2013), pp. 142–150.
- [130] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. "Image Plane Interaction Techniques in 3D Immersive Environments". In: *Proceedings of the Symposium on Interactive 3D Graphics*. ACM. 1997, pp. 39–43.
- [131] Wendy J. Plesniak. "Real-Time Computer Generated 3D Holograms". In: *Proceedings of SPIE*. 1993, pp. 536–543.
- [132] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. "The Go-go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR". In: *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. 1996, pp. 79–80.
- [133] Ivan Poupyrev, T Ichikawa, S Weghorst, and M Billinghurst. "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques". In: *Computer Graphics Forum*. Vol. 17. 3. Wiley Online Library, 1998, pp. 41–52.
- [134] Ivan Poupyrev and TADAO Ichikawa. "Manipulating Objects in Virtual Worlds: Categorization and Empirical Evaluation of Interaction Techniques". In: *Journal of Visual Languages & Computing* 10.1 (1999), pp. 19–35.
- [135] Ivan Poupyrev, Suzanne Weghorst, Takahiro Otsuka, and Tadao Ichikawa. "Amplifying Spatial Rotations in 3D Interfaces". In: *Proceedings of CHI Extended Abstracts on Human Factors in Computing Systems*. ACM. 1999, pp. 256–257.
- [136] Jenny Preece. *Interaction Design : Beyond Human-Computer Interaction*. West Sussex, 2011.
- [137] Erik Prytz, Michael Montano, and Mark W Scerbo. "Using Fitts' Law for a 3D Pointing Task on a 2D Display: Effects of Depth and Vantage Point". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 56. 1. SAGE Publications. 2012, pp. 1391–1395.

- [138] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. “Tilt Techniques: Investigating the Dexterity of Wrist-based Input”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1943–1952.
- [139] Jaime Ramirez. “Guidelines for Choosing VR Devices from Interaction Techniques”. In: *Proceedings of the 5th WSEAS International Conference on Multimedia, Internet & Video Technologies*. Citeseer. 2005, pp. 46–51.
- [140] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. “A Screen-Space Formulation for 2D and 3D Direct Manipulation”. In: *Proceeding of the ACM Symposium on User Interface Software and Technology*. 2009, pp. 69–78.
- [141] *RGBDSLAM*. 2015. URL: <https://openslam.org/rgbdslam.html>.
- [142] Meredith Ringel. “Barehands: Implement-Free Interaction with a Wall-Mounted Display”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2001, pp. 367–368.
- [143] David F. Rogers and Rae Earnshaw. *State of the Art in Computer Graphics: Aspects of Visualization*. 1st ed. Springer, 2014.
- [144] Jaime Ruiz, Yang Li, and Edward Lank. “User-defined Motion Gestures for Mobile Interaction”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 197–206.
- [145] Antti Sand, Ismo Rakkolainen, Poika Isokoski, Jari Kangas, Roope Raisamo, and Karri Palovuori. “Head-mounted Display with Mid-air Tactile Feedback”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. ACM. 2015, pp. 51–58.
- [146] Johannes Schöning, Frank Steinicke, Antonio Krüger, Klaus Hinrichs, and Dimitar Valkov. “Bimanual Interaction with Interscopic Multi-Touch Surfaces”. In: *Human-Computer Interaction – INTERACT 2009*. Vol. 5727. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 40–53.
- [147] Stacey D Scott, Karen D Grant, and Regan L Mandryk. “System Guidelines for Co-located, Collaborative Work on a Tabletop Display”. In: *Proceedings of ECSCW 2003*. Springer, pp. 159–178.

- [148] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. "System Guidelines for Co-located, Collaborative Work on a Tabletop Display". In: *Proceedings of the Eighth Conference on European Conference on Computer Supported Cooperative Work*. 2003, pp. 159–178.
- [149] Chia Shen, Frédéric D. Vernier, Clifton Forlines, and Meredith Ringel. "DiamondSpin: An Extensible Toolkit for Around-the-table Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2004, pp. 167–174.
- [150] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. "Motion Capture from Body-mounted Cameras". In: *ACM Trans. Graph.* 30.4 (July 2011), pp. 31–41.
- [151] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. "Real-time Human Pose Recognition in Parts from Single Depth Images". In: *Proceeding of CVPR*. 2011.
- [152] Linda E. Sibert and Robert J. K. Jacob. "Evaluation of Eye Gaze Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2000, pp. 281–288.
- [153] Sony SmartEyeglass. *Sony SmartEyeglass*. 2015. URL: <https://developer.sony.com/devices/mobile-accessories/smarteyeglass/>.
- [154] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P. Bailey, and Giuliano Macioccoi. "BeThere: 3D Mobile Collaboration with Spatial Input". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 179–188.
- [155] Peng Song, Woon Boon Goh, William Hutama, Chi-Wing Fu, and Xiaopei Liu. "A Handle Bar Metaphor for Virtual Object Manipulation with Mid-air Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 1297–1306.
- [156] S.E.B. Sorensen, P.S. Hansen, and N.L. Sorensen. *Method for Recording and Viewing Stereoscopic Images in Color using Multichrome Filters*. US Patent 6,687,003. 2004.
- [157] Oleg Špakov and Darius Miniotas. "On-line Adjustment of Dwell Time for Target Selection by Gaze". In: *Proceedings of the Nordic Conference on Human-computer Interaction*. ACM. 2004, pp. 203–206.

- [158] S.Pastoor. “Human Factors of 3D Imaging”. In: *HHI*. Berlin.
- [159] Olov Ståhl, Anders Wallberg, Jonas Söderberg, Jan Humble, Lennart E. Fahlén, Adrian Bullock, and Jenny Lundberg. “Information Exploration Using The Pond”. In: *Proceedings of the 4th International Conference on Collaborative Virtual Environments*. ACM, 2002, pp. 72–79.
- [160] Anthony Steed. “Towards a General Model for Selection in Virtual Environments”. In: *Proceeding of the IEEE Symposium on 3D User Interfaces*. IEEE. 2006, pp. 103–110.
- [161] Frank Steinicke, Klaus Hinrichs, Johannes Schöning, and Antonio Krüger. “Multi-Touching 3D Data: Towards Direct Interaction in Stereoscopic Display, Environments coupled with Mobile Devices”. In: *Proceedings of AVI PPD’08 Workshop*. 2008, pp. 46–49.
- [162] Richard Stoakley, Matthew J Conway, and Randy Pausch. “Virtual Reality on a WIM: Interactive Worlds in Miniature”. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 1995, pp. 265–272.
- [163] Sven Strothoff, Dimitar Valkov, and Klaus Hinrichs. “Triangle Cursor: Interactions with Objects Above the Tabletop”. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM. 2011, pp. 111–119.
- [164] S Subramanian and W IJsselsteijn. “Survey and Classification of Spatial Object Manipulation Techniques”. In: *Proceedings of OZCHI Interfacing Reality in the New Millennium*. 2000, pp. 330–337.
- [165] Bernard Suits. “What is a Game?” In: *Philosophy of Science* (1967), pp. 148–156.
- [166] Ivan E. Sutherland. “A Head-mounted Three Dimensional Display”. In: *Proceedings of Fall Joint Computer Conference*. ACM, 1968, pp. 757–764.
- [167] Haruo Takemura and Fumio Kishino. “Cooperative Work Environment Using Virtual Workspace”. In: *Proceedings of the ACM Conference on Computer-supported Cooperative Work*. ACM. 1992, pp. 226–232.

- [168] Yoshiki Takeoka, Takashi Miyaki, and Jun Rekimoto. “Z-touch: An Infrastructure for 3D Gesture Interaction in the Proximity of Tabletop Surfaces”. In: *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 2010, pp. 91–94.
- [169] John C. Tang. “Findings from Observational Studies of Collaborative Work”. In: *Int. J. Man-Mach. Stud.* 34.2 (1991), pp. 143–160.
- [170] Vildan Tanriverdi and Robert J. K. Jacob. “Interacting with Eye Movements in Virtual Environments”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2000, pp. 265–272.
- [171] Robert J Teather. “Comparing 2D and 3D Direct Manipulation Interfaces”. PhD thesis. York University Toronto, 2008.
- [172] *The International Conference on Mobile Computing and Networking*. ACM, 2014.
- [173] ENDO Tomohiro, KAJIKI Yoshihiro, HONDA Toshio, and SATO Makoto. “Cylindrical 3D Display Observable from All Directions”. In: *The Transactions of the Institute of Electronics, Information and Communication Engineers. D-II* 84.6 (2001-06-01), pp. 1003–1011.
- [174] Theophanis Tsandilas, Emmanuel Dubois, and Mathieu Raynal. “Free-space Pointing with Constrained Hand Movements”. In: *Proceedings of Extended Abstracts on Human Factors in Computing Systems*. ACM. 2010, pp. 3451–3456.
- [175] Unity. *Unity*. 2014. URL: <http://www.unity3d.com> (visited on 2014).
- [176] Hakan Urey, Kishore V Chellappan, Erdem Erden, and Phil Surman. “State of the Art in Stereoscopic and Autostereoscopic Displays”. In: *Proceedings of the IEEE* 99.4 (2011), pp. 540–555.
- [177] Dimitar Valkov, Frank Steinicke, Gerd Bruder, and Klaus Hinrichs. “2D Touching of 3D Stereoscopic Objects”. In: *In Proceedings of ACM Conference on Human Factors in Computing Systems*. 2011, pp. 1353–1362.
- [178] Dimitar Valkov, Frank Steinicke, Gerd Bruder, and Klaus Hinrichs. “2D Touching of 3D Stereoscopic Objects”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2011, pp. 1353–1362.

- [179] Janneke Van de Pol, Monique Volman, and Jos Beishuizen. “Scaffolding in Teacher–student Interaction: A Decade of Research”. In: *Educational Psychology Review* 22.3 (2010), pp. 271–296.
- [180] Jonathan Ventura, Clemens Arth, Gerhard Reitmayr, and Dieter Schmalstieg. “Global Localization from Monocular SLAM on a Mobile Phone”. In: *IEEE Trans. Vis. Comput. Graph.* 20.4 (2014), pp. 531–539.
- [181] *Virtual Surgery Table*. 2000. URL: <http://www.barco.com>.
- [182] Daniel Vogel and Ravin Balakrishnan. “Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM. 2005, pp. 33–42.
- [183] *Vuforia for Smart EyeWear*. 2015. URL: <https://www.qualcomm.com/%20products/vuforia>.
- [184] Robert Walter, Gilles Bailly, and Jörg Müller. “StrikeAPose: Revealing Mid-air Gestures on Public Displays”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 841–850.
- [185] R. Y. Wang and J. Popovic. “Real-time Hand-tracking with a Color Glove”. In: *ACM Transactions on Graphics (TOG)* 28.63 (2009), pp. 63–71.
- [186] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai. “Video-based Hand Manipulation Capture Through Composite Motion Control”. In: *Proceedings of SIGGRAPH*. 63. 2013, pp. 43–56.
- [187] Yanqing Wang and Christine L MacKenzie. “Effects of Orientation Disparity between Haptic and Graphic Displays of Objects in Virtual Environments”. In: *Proceedings of INTERACT*. 1999, pp. 391–398.
- [188] Yanqing Wang and Christine L. MacKenzie. “The Role of Contextual Haptic and Visual Constraints on Object Manipulation in Virtual Environments”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2000, pp. 532–539.
- [189] Colin Ware, Kevin Arthur, and Kellogg S. Booth. “Fish Tank Virtual Reality”. In: *Proceedings of the Conference on Human Factors in Computing Systems*. ACM, 1993, pp. 37–42.

- [190] Colin Ware and Kathy Lowther. "Selection Using a One-eyed Cursor in a Fish Tank VR Environment". In: *ACM Trans. Comput.-Hum. Interact.* 4.4 (Dec. 1997), pp. 309–322.
- [191] Colin Ware and Harutune H. Mikaelian. "An Evaluation of an Eye Tracker As a Device for Computer Input2". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems and Graphics Interface*. ACM. 1987, pp. 183–188.
- [192] Colin Ware and Steven Osborne. "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments". In: *Proceedings of the Symposium on Interactive 3D Graphics*. ACM, 1990, pp. 175–183.
- [193] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. "Analysis of the Accuracy and Robustness of the Leap Motion Controller". In: *Sensors* 13.5 (2013), pp. 6380–6393.
- [194] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. "Bringing Physics to the Surface". In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM. 2008, pp. 67–76.
- [195] C.A. Wingrave, B. Williamson, Paul D. Varcholik, Jeremy Rose, A. Miller, E. Charbonneau, Jared Bott, and J.J. LaViola. "The Wiimote and Beyond: Spatially Convenient Devices for 3D User Interfaces". In: *Computer Graphics and Applications, IEEE* 30.2 (Mar. 2010), pp. 71–85.
- [196] Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. "User-defined Gestures for Surface Computing". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1083–1092.
- [197] David Wood, Jerome S Bruner, and Gail Ross. "The Role of Tutoring in Problem Solving". In: *Journal of child psychology and psychiatry* 17.2 (1976), pp. 89–100.
- [198] Robert Sessions Woodworth. "Accuracy of Voluntary Movement". In: *The Psychological Review: Monograph Supplements* 3.3 (1899), pp. 1–119.
- [199] Robert Sessions Woodworth. "The Accuracy of Voluntary Movement". In: *American Journal of Psychiatry* 5.4 (1899), 128 pages.

- [200] Mike Wu and Ravin Balakrishnan. “Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. 2003, pp. 193–202.
- [201] H.P. Wyss, R. Blach, and M. Bues. “iSith - Intersection-based Spatial Interaction for Two Hands”. In: *Proceedings of the 3DUI*. IEEE. 2006, pp. 59–61.
- [202] N. Yang, M. Zhang, C. Huang, and D. Jin. “Motion Quality Evaluation of Upper Limb Target-Reaching Movements”. In: *Medical Engineering and Physics* 24.2 (2002), pp. 115–120.
- [203] Shunsuke Yoshida, Sumio Yano, and Hiroshi Ando. “Implementation of a Tabletop 3D Display Based On Light Field Reproduction”. In: *Proceedings of SIGGRAPH*. ACM. 2010, 1 page.
- [204] Lingyun Yu, Pjotr Svetachov, Petra Isenberg, Maarten H. Everts, and Tobias Isenberg. “FI3D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces”. In: *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1613–1622.
- [205] Shumin Zhai, William Buxton, and Paul Milgram. “The “Silk Cursor”: Investigating Transparency for 3D Target Acquisition”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 1994, pp. 459–464.
- [206] Pengfei Zhou, Mo Li, and Guobin Shen. “Use It Free: Instantly Knowing Your Phone Attitude”. In: *Proceedings of the International Conference on Mobile Computing and Networking*. ACM. 2014, pp. 605–616.