DISSERTATION

UNDERSTANDING USER INTERACTIONS IN STEREOSCOPIC HEAD-MOUNTED DISPLAYS

Submitted by

Adam S. Williams

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2022

Doctoral Committee:

Advisor: Francisco R. Ortega

Ross Beveridge
Joe Gersch
Julia Sharp

ABSTRACT

UNDERSTANDING USER INTERACTIONS IN STEREOSCOPIC HEAD-MOUNTED DISPLAYS

Interacting in stereoscopic head mounted displays can be difficult. There are not yet clear standards for how interactions in these environments should be performed. In virtual reality there are a number of well designed interaction techniques; however, augmented reality interaction techniques still need to be improved before they can be easily used. This dissertation covers work done towards understanding how users navigate and interact with virtual environments that are displayed in stereoscopic head-mounted displays. With this understanding, existing techniques from virtual reality devices can be transferred to augmented reality where appropriate, and where that is not the case, new interaction techniques can be developed. This work begins by observing how participants interact with virtual content using gesture alone, speech alone, and the combination of gesture+speech during a basic object manipulation task in augmented reality. Later, a complex 3-dimensional data-exploration environment is developed and refined. That environment is capable of being used in both augmented reality (AR) and virtual reality (VR), either asynchronously or simultaneously. The process of iteratively designing that system and the design choices made during its implementation are provided for future researchers working on complex systems. This dissertation concludes with a comparison of user interactions and navigation in that complex environment when using either an augmented or virtual reality display. That comparison contributes new knowledge on how people perform object manipulations between the two devices.

When viewing 3D visualizations, users will need to feel able to navigate the environment. Without careful attention to proper interaction technique design, people may struggle to use the developed system. These struggles may range from a system that is uncomfortable and not fit for long-term use, or they could be as major as causing new users to not being able to interact in these

environments at all. Getting the interactions right for AR and VR environments is a step towards facilitating their widespread acceptance. This dissertation provides the groundwork needed to start designing interaction techniques around how people utilize their personal space, virtual space, body, tools, and feedback systems.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Francisco R. Ortega, for his support, encouragement, and mentorship. I also would like to thank my Ph.D. committee members, Dr. J. Ross Beveridge, Dr. Joe Gersch, and Dr. Julia Sharp for offering their time and advice.

I'd like to acknowledge all of the members of the Natural User Interaction Lab members who have helped throughout this research. In particular, I would like to thank Xiaoyan Zhou for helping me run and interpret studies and Aditya Raikwar for his help with writing code and designing components of the system.

We would also like to thank our sponsors for there support. This work was supported by the Wim Bohm and Partners PhD Support award, the National Science Foundation (NSF) awards: NSF-2106590, NSF-2016714, NSF-2037417, NSF-1948254, NSF-192850, and the Defense Advanced Research Projects Agency (DARPA) award number W911NF-15-1-0459.

Thank you to Logitech for the Logitech VR INK Pilot Edition pen.

DEDICATION

This dissertation is dedicated to my partner, Sarah, my advisor Francisco, and my mentor Cap. This would have have been a very different journey without their support. Cap included me in research that he was a part of in the psychology department. During this time I was introduced to research, experiment design, web development, and the types of critical thinking needed to conduct research. I would like to thank Cap for that inclusion and introduction to the research community. That inclusion and mentor-ship ultimately led me to meet Francisco when he began working at Colorado State University. Francisco took a chance on accepting me as a student. I represented a risk because I was coming to computer science from the business school. At every step of the way Francisco was there to provide any support that was needed, putting me in a position to succeed during this degree program. I am very grateful for all that he has done for me. To Sarah, I don't think that the last few years of the program would have been the same without you. Thank you for your endless patience, support, and understanding during this process.

LIST OF WORKS RELEVANT TO THIS DEGREE

Refereed Journals

Williams, A. S., Garcia, J., De Zayas, F., Hernandez, F. Sharp, J., and Ortega, F. (2020). "The Cost of Production in Elicitation Studies and the Legacy Bias-Consensus Trade off". *Multimodal Technologies and Interaction, 4, 88. DOI: https://doi.org/10.3390/mti4040088*

Williams, A. S., *Ortega, F. (2020). "Understanding Gesture and Speech Multimodal Interactions for Manipulation Tasks in Augmented Reality Using Unconstrained Elicitation". Proc. ACM Human-Computer Interaction. V4, ISS, Article 202 (November 2020), 21 pages. DOI: https://doi.org/10.1145/3427330*

Williams, A. S., *Garcia, J., Ortega, F. (2020). "Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation". in IEEE Transactions on Visualization and Computer Graphics, DOI: https://doi.org/10.1109/TVCG.2020.3023566, Impact Factor: 4.56, Acceptance Rate: 6%*

Refereed Workshop Articles

Williams, A., *Ortega, F. (2021), "Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment". In IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW),2021, pp. 205-209, DOI: https://doi:10.1109/VRW52623.2021.00045*

Williams, A., *Ortega, F. (2020), "Multimodal User-Defined inputs for Optical See Through Augmented Reality Environments". In IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pp. 557-558, DOI: https://doi.org/10.1109/VRW50115.2020.00130*

Williams, A.S., *Ortega, F. (2020), "Conversations On Multimodal Input Design With Older Adults," CHI 2020 (Designing Interactions for the Ageing Populations – Addressing Global Challenges), Honolulu, Hawaii, 2020, https://arxiv.org/abs/2008.11834*

Williams, A.S., *Ortega, F. (2020), "Insights on visual aid and study design for gesture interaction in limited sensor range Augmented Reality devices," In IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), pp. 19-22, 2020, DOI: https://doi.org/10.1109/VRW50115.2020.00286*

Ortega, F., Kress, M., Tarre, K., **Williams, A.**, *Rishe, N., and Barreto, A. (2019), "Selection and Manipulation Whole-Body Gesture Elicitation Study in Virtual Reality," In IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (NIDIT), Osaka, Japan, 2019, pp. 1110-1111. DOI: https://doi.org/10.1109/VR.2019.8798182 - Short paper*

Ortega, F., Kress, M., Tarre, K., **Williams, A.**, *Rishe, N., and Barreto, A. (2019), "Selection and Manipulation Whole-Body Gesture Elicitation Study In Virtual Reality," In IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (NIDIT), Osaka, Japan, 2019, pp. 1723-1728. DOI: https://doi.org/10.1109/VR.2019.8798105 - Workshop Paper*

Books

***Williams, A. S.**, & Ortega, F. R. (2021). A Concise Guide to Elicitation Methodology. arXiv e-prints, arXiv-2105. https://arxiv.org/abs/2105.12865*

Magazine Articles

***Williams, A.S.**, and Ortega, F.R. (2020) "Evolutionary gestures: When a gesture is not quite legacy biased". In ACM interactions 28, 4 (October - September 2020), DOI: https://doi.org/10.1145/3412499*

Courses

Williams, A.**, and Ortega, F.. (2022). "An Introduction to Elicitation Study Design ". In Human Computer Interaction International (HCII 2022). Human Computer Interaction International, Virtual, USA, **(Upcoming: 06/26/22 - 07/01/22)

*Ortega, F., **Williams, A.**, and Garcia, J. (2020). "Multi-modal gesture elicitation methodology for children". In Proceedings of the 2020 ACM Interaction Design and Children Conference:*

Extended Abstracts (IDC '20). Association for Computing Machinery, New York, NY, USA, pp. 85–88. DOI: https://doi.org/10.1145/3397617.3401808

Masters Degree Thesis

Williams, A. *(2020). "The Impact of Referent Display on Interaction Proposals During Multimodal Elicitation Studies". Colorado State University. https://hdl.handle.net/10217/232528,* ***(Embargo expiration date: 06/02/2022)***

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

This dissertation covers research done to improve our understanding of user interactions in AR environments. This course of research begins with participatory design studies examining multimodal inputs in basic environments rendered in augmented reality (AR) head-mounted displays (HMDs) and ends with an examination of user behaviors in a complex environment, comparing across AR and virtual reality (VR) HMDs.

Data is being collected in unprecedented quantities and AR is quickly becoming consumer-available. Representations of data are more digestible and interpretable to users if they are easy to interact with and immersive. AR and VR can both afford those experiences. If people can intuitively explore data to draw their own conclusions, they will have more faith in those conclusions. For people to efficiently navigate and interact with 3-dimensional (3D) stereoscopic visualizations, the interface must be invisible. Little work has examined what interaction techniques are appropriate for use with AR-HMDs. Natural user interactions, i.e., speech, gesture, multimodal combinations of speech and gesture, pen, and gaze, can make that interface invisible. Without intuitive interaction techniques, these emerging systems will suffer from low immersion and user flow, degrading user's work performance.

1.1 Background

AR devices are becoming increasingly available. The Microsoft Hololens 2, an optical see-through stereoscopic AR-HMD, has already established a 480 million dollar contract to sell as many as $100,000$ devices to the United States Army [3]. While currently targeted at enterprise use, shortly AR technologies will be in the hands of the public. The long-term vision of AR technologists is to have devices with form factors similar to glasses that will take the place of consumer cell phones, or augment phone use as smartwatches have. AR technology continues

to improve, with Microsoft Hololens 3 in development and rumored AR product launches from Facebook and Apple as early as 2021 [1].

Alongside AR-HMDs, VR-HMDs are rapidly advancing. Some current generation VR-HMDs can be purchased for less than $400, half of the first generation HTC Vive VR-HMD. VR-HMDs offer a different experience than AR-HMDs. When in a VR-HMD, the outside world is occluded by the displays used. This means that the user is more fully immersed in the virtual environment being presented to them. VR-HMDs also typically have better resolutions, fields of view, and controllers than AR-HMDs.

Outside of headsets, the rate that people generate and store data is another rapidly growing area of society. Data collection is occurring at unprecedented levels [4], with nearly $2,500,000$ Terabytes of data produced globally every day [5, 6]. Sources of this data are oftentimes low-cost and nearly ubiquitous sensors found in smartphones, HMDs, watches, online consumer activity, news reports, or even corporate reports on various sales and cost metrics. Interpretation of this data can allow users to develop a wider knowledge base of many things from personal health tracking to the accuracy of news reports.

With these areas of society each evolving, there is a gap in the literature and a growing need to answer the question; what is the best way to interact with this rapidly growing data? AR-HMDs will often leverage gesture and speech commands as seen in the Microsoft HoloLens 2 and the Magic Leap One. These natural interfaces are easy to carry and if done right, intuitive. Interfaces that are easy to use can lower the novice to expert skill gap [7] and are more discoverable to end-users [8]. Similarly, complex data is more digestible when users can interact with it easily [4]. While these input modalities are being shipped with AR-HMDs, there is not a clear standard for input design in AR [9], or for three-dimensional (3D) data interactions [10, 11].

VR-HMDs have been more commonly utilized for researching interactions with 3D data displays, however; research on interaction techniques for VR data visualization environments is lack-

[1] https://www.tomsguide.com/news/apple-glasses,https://tech.fb.com/facebook-connect-the-road-to-ar-glasses/

ing [10]. Additionally, the impact of using a VR-HMD for 3D data exploration compared to an AR-HMD is unknown.

This work rests at the intersection of data visualization, augmented reality, virtual reality, and user interaction techniques. When data visualizations are displayed in interactive and immersive 3D environments, it is referred to as "Immersive Analytics" (IA) [10, Chapter 1]. IA is a young field, having come into the public eye at a workshop in 2015 [12]. The same workshop led to the publication of the IA textbook in 2018 [10].

1.1.1 Motivation

The cost of VR and AR head-mounted displays is dropping while their quality is increasing. AR-HMDs may soon become pervasive and yet, there is currently very little work on interactions in AR-HMDs [9], and even less on interactions in AR-IA [10, 11, Chapter 4]. Society's most recent large shift in interactive technologies was the transfer from 2D computers to 2D multi-touch cellular phones, devices with somewhat similar displays and interaction techniques. Both technologies used 2D screens and arguably, the interactions done with a mouse (i.e., selection, clicking, dragging) can be well achieved with multi-touch. The shift from 2D interactions and displays to stereoscopic displays and 3D interaction techniques poses a larger technological leap from an interaction design standpoint.

Alongside AR and VR's increase in availability, there is a shifting landscape in the ways that office workers can work. COVID-19 has transitioned many traditionally office-based jobs to remote jobs, creating a need for society to adapt to working remotely. Even after COVID-19 is no longer an international concern, remote work may be here to stay. Many companies have already announced plans to continue remote work indefinitely [13–15]. Both employees and employers can benefit from the traditional work to remote work transition. If employees can work from their homes, then there is no overhead for maintaining an office building. The costs saved range from heating and cooling, septic systems, and building insurance to facilities management including

parking-lot maintenance. The employee may also advocate for remaining at home due to lower commute times and less cost in travel.

This new landscape of remote work and personal stereoscopic use makes it necessary to develop interaction techniques that allow employees to effectively work remotely. AR and VR displays offer a viewing environment with "infinite pixels" in comparison to a monitor where there is a set amount of render-able surface. In AR and VR, the user can render many monitors across their office for the cost of a single device (e.g., AR-HMDs, VR-HMDs). The immersive environments in these headsets can also allow for collaborative work environments with some work showing that IA improves collaboration compared to standard co-located collaboration [16]. These display devices each provide different advantages and may be best suited for different tasks. Before we can develop interaction techniques for IA environments using stereoscopic devices we need to understand how differences in the displays used by these headsets impacts user interactions within them.

Society as a whole will also face this shift in data interpretation. Soon people will have personal AR and/or VR HMDs and an abundance of health tracking data. Schools may teach using these displays and children will need to interact with visualizations in their course work. These interaction techniques can be used for information displays beyond just graphics; people could be building molecules [17], or learning anatomy with a rendered body [18].

Poorly developed interaction sets pose another barrier for the adoption of this new system of work. Unintuitive or difficult-to-use interactions can slow users down, break immersion, and cause high levels of frustration. There is little work on appropriate interactions for IA and many constraints. Some users may need to interact in an IA environment over long periods of time [19] while others might be attending a virtual meeting that is an hour-long or interacting with a small chemistry assignment dataset for only a few minutes. A shift worker may be set up in an office and need controls that can support long-term use (i.e., speech, multi-touch, micro-gesture, keyboard). A person attending the meeting may be more willing to use mid-air gestures due to the low risk of fatigue in inconsistent and intermittent use. The person interacting with the chemistry data may be

traveling on a subway making speech and keyboard controls difficult. To facilitate these diverse use cases, we must develop robust, intuitive, and varied interaction techniques that can utilize unimodal and multimodal input technologies.

Before we can develop these inputs we need to understand the user, their input preferences, and the constraints of these inputs use. As of now, there have been very few works in IA examining the user, their styles of input, and their preferences of input. This work represents a holistic approach to IA interaction design that considers the range of use constraints and utilizes a collection of technologies to facilitate a robust exploration of interaction techniques across complex stereoscopic environments.

1.2 Previous Work

This work draws upon multimodal interaction design from Human-Computer Interaction (HCI) and more generic AR/VR environments. This interaction knowledge will be used alongside the limited existing work on IA and information visualization interaction techniques.

1.2.1 Multimodal Interaction

Outside of interaction techniques, IA systems can also provide implement new tools for the analysis and interpretation of data. The ability to annotate information while actively reading is an important component of data exploration. Annotations can deepen learning, understanding, and increase the exploration of data [20]. Linking annotations to the underlying data structure also serves an important role in sense-making [20, 21]. An example of one possible annotation tool is writing, which has many advantages over typing in terms of memory and retention [22]. Pen use has been examined in the context of paper [23], multi-touch surfaces [24], bi-manual interactions [24, 25], virtual reality [26, 27], and multi-touch surface annotation [20], but no examination has assessed how users will annotate visualizations with mid-air pen and speech in augmented reality (AR) environments.

AR technologies are starting to become available to knowledge workers and consumers. With this technology's emergence, now is the time to explore mid-air pen use in optical see-through AR. The space inside optical see-through Augmented Reality headsets is unique. It removes many of the affordances found with analog pen and paper as well as those found in digital pen and paper. In AR, there is no obvious surface to write on. The white space provided by paper is a useful tool for data exploration, often being used to provide a space for annotations [28]. White space use is complex and understanding its manipulation has been difficult [28]. Techniques for pen annotation may not carry over well when there is no passive kinesthetic feedback, or the affordances and white space found in paper.

1.3 Dissertation Layout

The path towards understanding how these multimodal inputs should be implemented is a complex one. AR displays are only beginning to become consumer-available and immersive analytics environments are still primarily used in research settings. To understand how users interact in complex environments we need to first build up knowledge of user behaviors and interactions in basic AR environments.

This dissertation follows that path, starting with two multimodal interaction studies done in AR (Chapter 2) [29, 30]. One of these studies focuses on observing user gesture interactions and collecting speech and gesture+speech interaction information to compare with the second study, while the other study focuses on user speech and gesture+speech interactions, collecting gesture interaction information to compare against the first study's elicited gesture interactions. The comparison of these studies is not presented here but was submitted as my master's thesis.

With the knowledge gained from those studies, we turned to researching how interactions are made with complex environments such as those used in IA. Transitioning from the basic environment to a more complex one led to the development of a cross-device IA environment with markup tools and multimodal inputs. This system was developed over several iterative design sessions and

is one of the first of its kind. The development of this system and the design choices made while creating it are covered in Chapter 3.

Once developed, four pilot studies were run using this system in preparation for running the full study. The results of these pilots suggested that jumping from a basic environment in AR to a new complex IA environment required more work establishing how users interact with the objects in the system. This system provided a rich interaction space in which researchers could observe participants execute a range of actions from writing in mid-air with a pen to highlighting a region of a 3D scatterplot. When examining user interactions in this environment, we found that the system may be too unfamiliar or novel to conduct such a study. Some indications of that were that some participants struggled to perform even basic interactions in this environment. In response to that observation, we instead ran a between-groups observational study to investigate how users interact in this unfamiliar IA environment and how the choice of stereoscopic display impacts those interactions. This study compared AR and VR user behaviors and interactions while using this IA system in a general data exploration task. Knowing the differences between AR and VR behaviors allows researchers to know how to leverage each device's strengths. This comparison and its discussion are presented in Chapter 4. Finally, this dissertation concludes with the next steps for the environment and research in this area (Chapter 5).

1.4 The Format of This Dissertation

This dissertation is a cumulative collection of some of the works completed over the course of my studies on user interactions. Works that are relevant to the main goals of this dissertation research topic are presented as they were published in top-tier peer-reviewed venues. These works are introduced and concluded with connections to why they were a necessary step taken during this research. Including these publications gives important insights into why key decisions were made during this course of research.

Not all works done during my degree are presented here, some works are omitted due to divergent topics. An example of this is seen with a paper that was presented at the Association for

Computing Machinery (ACM) CHI "Designing Interactions for the Ageing Populations Addressing Global Challenges" workshop that looks into what types of inputs older adults might benefit from in VR [31]. Another such work is the textbook that was written in conjunction with my advisor Dr. Francisco Ortega that guides researchers through the process of running and interpreting elicitation studies [32]. This book was based on the experiences gained during the two elicitation papers presented here and a course Francisco and I taught at the 2020 ACM Interaction Design and Children Conference [29, 30, 33]. A tutorial on a similar topic will be presented at the upcoming 2022 Human-Computer Interaction International Conference. Works that were done while completing this degree but were not directly relevant to this dissertation are listed in Appendix A.1.

1.5 Contributions

Chapter 2 provides findings on how people interact using gestures alone, speech alone, and gesture+speech together, in basic environments that are rendered on AR-HMDs. These findings include a set of discoverable gesture interactions, a set of common hand poses, the most prevalent speech commands used, time information for the co-occurring gesture and speech interactions, and the differences in perceived workload between the input modalities used.

Chapter 3 then discusses the iterative design and implementation of a complex immersive analytics platform. This platform can support remote and co-located collaboration, asynchronous use, six annotation tools, and can be run on a PC, VR-HMD, or AR-HMD. This platform is one of the first of its kind and the platform along with the decisions made as a result of the iterative design sessions are contributions of this work.

Differences in how AR-HMD and VR-HMD users navigate and interact within that complex environment are detailed to contribute new knowledge of how those two stereoscopic devices influenced participants' use of the system (Chapter 4). Contributions of Chapter 4 include knowledge on how users perform rotations, translations, and scaling inside of IA environments and how users manage both their virtual and physical space while wearing these devices.

Chapter 2

Multimodal Elicitation Studies

The first goal of this course of research was to develop a better understanding of how people interact using gesture alone, speech alone, and gesture+speech combined as input modalities in AR object manipulation environments. Two studies examining user interactions and behaviors in AR were conducted towards that goal. These two studies were very similar apart from the presentation of the referent (e.g., the command that interactions are being elicited for). Both studies are presented in this chapter as they were published [29, 30]. These articles are included with copyright holder permission. This chapter begins by providing additional information about those studies and concludes with where the results of those studies led this research.

2.1 Research Aims

The aim of these works was to find how people interacted in basic AR environments using multimodal inputs. The input modalities used were gesture alone, speech alone, and the combination of gesture+speech. These input modalities were chosen based on their intuitive nature and ease of access. The exact environments that AR-HMDs will be used in are unknown. Currently, AR-HMDs are largely used in industry [3, 34]. Conceivably, office workers and everyday citizens will use AR-HMDs during their daily activities. Evidence of this possibility is seen in the assortment of AR glasses currently under development or already on the market[2]. Gestures and speech are the base case input modality that will likely be available to users in a variety of environments and are the inputs currently shipped with most AR-HMDs.

[2]tomsguide.com/news/apple-glasses, tech.fb.com/facebook-connect-the-road-to-ar-glasses

2.2 Extended Study Details

This section provides more granular details on the design of the two studies, recruitment of participants, data collection, data preparation, and the analyses performed.

2.2.1 Study Design

The two experiments presented here were both Wizard-of-Oz (WoZ) design elicitation studies. This means that participants interacted with the AR system while the experimenter acted as a recognizer for their inputs. This way their inputs were unconstrained by the systems recognition capability, allowing observation of what these participants would intuitively choose to use when interacting in AR.

These participants' interactions were limited to the modality condition that they were currently in, i.e., if the modality condition was gesture then they were asked to only use gesture-based interactions. For each input modality participants were asked to produce any interaction they felt was appropriate for executing the command presented to them. Seventeen commands, also called referents, were used. These referents covered the canonical interactions done when manipulating objects in 3D environments.

At the beginning of a session, participants arrived at a lab hosted by the university and completed an informed consent form and a demographics questionnaire. The demographics questionnaire inquired about participant's prior exposure to mid-air gesture systems, VR-HMDs, and AR-HMDs, and demographic information such as gender, age, and major.

Prior to putting on the AR-HMD participants watched video instructions explaining the experiment and what was expected of them. There were two versions of these instructions, one for each experiment. Once in the environment, participants performed one practice trial for each input modality. During this practice, participants were asked to produce an input that would change a virtual cube's color. Next, the three input modalities were presented in counterbalanced order with the referents being randomized within each modality. With this design, participants were presented with an input modality (i.e., gesture), and then asked to produce an input proposal for each referent

using that input modality. Once each referent had been completed for that input modality, participants filled out a NASA-TLX survey for it. This was done to assess their perceived workload for using that input modality [35]. Following the NASA TLX, the next input modality condition was completed. This process ended when all input modalities were completed.

All interactions in this environment were performed on a virtual cube that was rendered about 50 cm in front of a user's head. A cube was selected because of its simple shape whose simplicity can help lessen the impact that the shape of an object has on a user's proposed interactions [36].

2.2.2 Methods

In the first study, the referents were read aloud and presented as text on the AR-HMD, and participants were told that their interactions were being recognized by the system [29]. Once a participant proposed an interaction for the given referent, an animation of that referent being executed was played. This was done to create a greater sense of engagement for the participants where they could feel like they were interacting with a live system. These animations lasted for 2 seconds after which a blue screen was shown. Once ready, they progressed to the next referent, and the blue screen was removed.

In the second study, the referents were presented as animations of the referent without the accompanying text or spoken instructions [30]. These animations were as similar to the animations played after the interaction proposals in the first experiment as possible. Participants were told that they were guessing what interaction a user in another room did to cause the animation shown. This provided a sense of engagement similar to the first experiment. Blue screens were shown between referents in this study as well.

A number of studies have used text referents when eliciting interaction proposals [36–40] while a number of other studies have used animations [30, 41–45]. Most of these studies were unimodal elicitation studies meaning the best mechanism for referent display when eliciting multimodal inputs was unknown. Gestures can be primed by animations where speech can be primed by text or spoken word. By running two studies, we were able to mitigate some of the impacts of

this priming to better compare the two referent displays. This comparison allowed us to make recommendations for future multimodal elicitation studies based on the impact that referent display had on the interaction proposals.

Referents Used

Referents were selected to capture the canonical manipulations performed in 3D environments [46, 47]. These canonical referents were translation on each axis (i.e., x, y, z), rotation about each axis, and scaling an object to be larger or smaller. Three additional referents were added to increase the generalizability of the results. These were creating an object, destroying an object, and selecting an object. For all referents apart from select, participants were told that the cube shown to them was already selected. The same set of referents was used for both studies.

Pilot Studies

One survey, two pilot studies, and one observational session were run prior to conducting the two studies presented here. No one participant was included in more than a single study. The survey was done to assess the general understanding of the terms that were being used for the referents (i.e., yaw right). This survey was administered to 35 people, all of whom were in an entry-level computer science course. Additionally, both versions of the final study were run as pilot studies with 6 participants each to help determine the level of biasing present in the different methods of referent display.

Prior to running the second study where the referents were presented as animations, an observational study was conducted with 5 participants. These participants were shown the animations that would be used in the full study along with some additional animations for the more abstract referents (i.e., create, destroy, select) and asked what they thought the animations were of. This was done to ensure that participants were perceiving the animations as the commands that were being elicited. The results of this observational study revealed that referents not grounded in real-world physics were difficult to animate. These referents were select, create, and destroy. For those

difficult to animate referents, the animations with most participants interpreting them correctly were selected.

Participants

Participants for these studies were all recruited using word of mouth advertising and department-wide emails. Participants were all either volunteers or given course credit for their participation. More details on the participants are given in the papers for these studies. Each study used a sample size of 24 which was grounded in prior work in the field, best summarized by a recent literature survey Of 216 elicitation studies [48]. This survey found that the mean participant count for elicitation studies was 25 with a standard deviation of 4 [48]. After allowing for a 3 condition counterbalanced design, the sample size of 24 was chosen.

Data Collection

Both ego-centric and exo-centric video was recorded of the participants interacting in this system. Videos were captured using the on-device ego-centeric camera, a Go-Pro camera that was worn above participant's AR-HMD, and an exo-centric camera that faced participants. The session audio was captured with these videos. The demographics survey and NASA TLX were collected online and saved as .csv files. This process generated one demographics survey, 3 videos, and 3 NASA TLX responses per participant.

Data Preparation

The pilot survey data was merged across participants to generate a .csv file with one column per referent and one row per participant. The cells in this .csv listed a binary variable indicating if the participant defined that referent correctly or not. The videos collected were watched by the experimenter while the experimenter took notes on what indications of referent priming were present in the elicited interaction proposals. These notes would include information like which referents a participant repeated in the speech condition or which animations participant's gestures imitated in the gesture condition.

The video from the observation sessions was watched and annotated to generate a file with one column per participant and one row per animation. The cells in this csv indicated if the participant correctly identified the animation. Animations that were not correctly identified were given a note saying what the animation was identified as.

The Go-Pro video footage was hand-annotated by the researcher using Microsoft excel. The other video footage was used as a fallback if the go-pro video was unusable for a given referent. This occurred most often when a participant's hands went out of frame. This process resulted in a file with annotations for each participant's 17 gesture proposals + 17 speech proposals + 17 gesture+speech proposals. These annotations included the type of gesture performed, fingers used, direction of movement, or words uttered in the case of speech.

These annotated proposals were then binned into equivalence classes. This process was slightly different for each input modality. The gesture proposals were combined into equivalence classes based on their similarity of execution. Proposals with similar motions and hand shapes were combined into one class. An example class is "finger-based x-axis pushing gesture" that would contain both two finger pushing gestures and one finger pushing gestures. This class would not include hand pushing gestures where four or more fingers were used. These classes use reversible directions for rotations and translations meaning that a push left with an open hand and push right with an open hand were combined into a single "open hand pushing along the x-axis gesture" class.

Speech proposals were binned based on the syntax used and phrases used. The syntax binning process broke speech proposals into action words, direction words, object specifiers, and other. The full phrases were binned based on the similarity of words used as seen with the gesture proposals. During this binning process "move backwards" and "move backward" were considered the same but "move back" was considered different.

Gesture+speech proposals were broken into gesture proposals and speech proposals then binned using the processes done for gesture only and speech only proposals. Gesture+speech interaction time stamps were collected from the raw videos. Timestamps were recorded for when a gesture started when speech started, when a gesture's stroke or first major change in direction occurred

when the speech was concluded, and when the gesture was concluded. The times between gesture initiation and speech initiation were then calculated such that a negative time meant that speech was started before the gesture was started. This process resulted in a .csv file with columns for each time stamp collected plus one for the time difference between gesture and speech initiation. This file had 17 rows for each participant (one row per referent).

The individual .csv files for all participants were merged into a single file for analysis resulting in one file for the demographics survey and one file for each input modalities NASA TLX responses. The demographic data included information on prior device use information, age, gender, eye-sight, and major or job type information.

Analysis Performed

Pilot survey data was analyzed using the raw counts of correctly or incorrectly defined referents. This provided information on which referent terms would be most likely understood by participants. Similarly, the raw counts of the correctly and incorrectly identified animations during the observational sessions were used to determine which animations were used in the animated referent study.

Gesture Metrics Agreement Rate ($\mathcal{AR}$) was used to calculate the level of participant agreement on gesture interaction proposals for each referent. Equation 2.1 shows the agreement rate formula. When using a sample size of 24, an $\mathcal{AR}$ of 0.3 is considered high agreement [49], with high agreement suggesting that novice users of a system would be able to discover the most commonly proposed interaction for that referent. In Equation 2.1, P is the set of all proposals for referent r, and P_i are the subsets of equivalent proposals from P [49].

$$\mathcal{AR}_r = \frac{\sum_{P_i \subseteq P} \frac{1}{2}|P_i|(|P_i| - 1)}{\frac{1}{2}|P|(|P| - 1)} \tag{2.1}$$

Fleiss' Kappa Fleiss' Kappa was used to calculate the level of change agreement (P_e) within the elicited proposals as suggested by Tsandilas (2019) [50]. The formula used is shown in Equa-

tion 2.2. In that equation m is the total number of proposals. The term n_{ik} represents the number of participants proposing interaction proposals i for bin k. The term π_k represents the chance that a rater classifies a proposal into bin k based on the times bin k was used across all proposals. The term q represents the total possible space of proposals. This formula is used to determine if the $\mathcal{AR}$ metric is a reasonable measure of participant agreement or if the $\mathcal{AR}$ was inflated by chance agreement between proposals.

$$p_e = \sum_{k=1}^{q_-} \pi_k^2, \quad \pi_k = \frac{1}{m} \sum_{i=1}^{m} \frac{n_{ik}}{n_i} \tag{2.2}$$

Speech Analysis The metrics used for speech analysis were max-consensus ($\mathcal{MC}$) and the consensus -distinct ratio ($\mathcal{CDR}$). $\mathcal{MC}$ is the percent of participants proposing the most common utterance proposal [51] and $\mathcal{CDR}$ is the percent of proposals for a referent that has over a baseline of 1 participants proposing them [51]. These two metrics are used to indicate the most common proposal in the data and the spread of proposals across the data. A high $\mathcal{MC}$ indicates high agreement between participants on a single proposal where a high $\mathcal{CDR}$ indicated that many proposals were agreed upon by 2 or more participants. Speech proposals were also analyzed using the binned syntax's rate of use. This analysis was done to identify the common speech proposal structures used.

Time Windows Analysis The times between gesture and speech initiations in the gesture+speech block were analyzed using Wilcoxon rank-sum tests as informed by Shapiro-Wilk tests of normality. The Wilcoxon rank-sum tests were used to determine the median time between gesture and speech initiation. The goal of the Wilcoxon rank-sum test was to find what time windows could be reasonably expected when developing gesture+speech recognition systems.

NASA TLX Analysis Means and standard deviations were computed for NASA TLX results. Differences between the results for different input modalities were compared using Welch Two Sample T-Tests after running Shapiro-Wilk tests for normality.

Consensus Set When consensus sets were provided they were based on the most frequent gesture proposal for the gesture and gesture+speech conditions. Recommendations for the use of the consensus sets were based on the $\mathcal{AR}$ rates computed for each referent.

2.3 Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation[3]

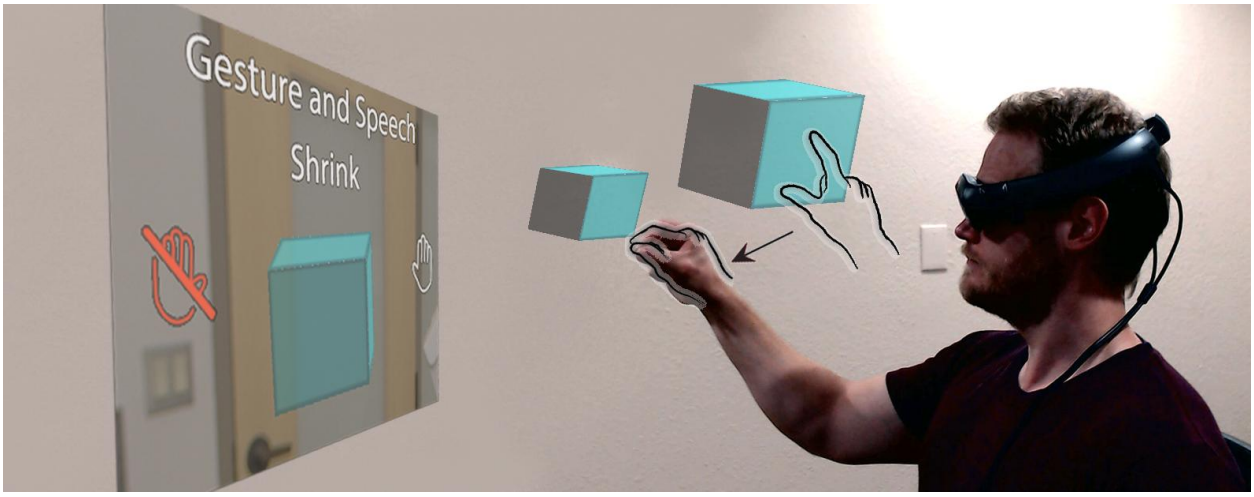


Figure 2.1: Example of experiment design: Left: participant view, Middle (hand outlines): gesture used, Right: Participant

2.3.1 Overview

The primary objective of this research is to understand how users manipulate virtual objects in augmented reality using multimodal interaction (gesture and speech) and unimodal interaction (gesture). Through this understanding, natural-feeling interactions can be designed for this technology. These findings are derived from an elicitation study employing Wizard of Oz design aimed

[3]Williams, A. S., Garcia, J., Ortega, F. (2020). "Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation". in *IEEE Transactions on Visualization and Computer Graphics, DOI: https://doi.org/10.1109/TVCG.2020.3023566*

at developing user-defined multimodal interaction sets for building tasks in 3D environments using optical see-through augmented reality headsets. The modalities tested are gesture and speech combined, gesture only, and speech only. The study was conducted with 24 participants. The canonical referents for translation, rotation, and scale were used along with some abstract referents (create, destroy, and select). A consensus set of gestures for interactions is provided. Findings include the types of gestures performed, the timing between co-occurring gestures and speech (130 milliseconds), perceived workload by modality (using NASA TLX), and design guidelines arising from this study. Multimodal interaction, in particular gesture and speech interactions for augmented reality headsets, are essential as this technology becomes the future of interactive computing. It is possible that in the near future, augmented reality glasses will become pervasive.

2.3.2 Introduction

Understanding multimodal interaction within augmented reality (AR) head-mounted displays (HMDs) is an important step towards improving user interactions. When used as unimodal inputs gestures and speech each have their strengths [51]. Gestures can be beneficial for direct manipulation of virtual objects where speech can be beneficial for abstract tasks such as creating new objects. The combination of gesture and speech, abundant in everyday life, can provide richer information than using either of those modalities alone. The synergies and individual merits of these modalities have not yet been fully examined in AR-HMD environments. Consider the impact that the desktop computer, smartphone, and tablet have had on people's lives. Augmented reality is one of the key technologies expected to have similar impacts on people's lives. As such, understanding the best inputs and combinations of inputs for use in this emerging technology is necessary. Unlike multi-touch devices, as of now, there exists no clear standard when it comes to mid-air gestures for use in AR environments [52].

The primary objective of this research is to understand how people naturally manipulate virtual objects in AR environments using multimodal interactions (gesture and speech) and unimodal interactions (gesture). This is done by observing participants perform these interactions in an un-

constrained environment. All inputs within each modality were accepted (i.e. any mid-air gesture or utterance). Given the nature of combining gesture with speech, speech alone was also examined. This addition allowed for a better analysis of how speech is formed with and without gestures. A secondary goal of this research is to assist in understanding how existing knowledge about gesture and speech interactions from psychology [53–55] hold once technology (in particular, AR) is added to the equation. Thus helping bridge the existing knowledge on human to human communication with human to computer communication.

End users represent a broad range of preferences. While most users prefer multimodal gesture and speech interactions, some users will prefer speech alone, or gesture alone [56]. With these varying individual preferences implementing gesture and speech alone as well as combined is important.

Contributions

The main contributions of this paper are:

1. A novel within-subjects multimodal and unimodal elicitation study for object manipulation tasks in optical see-through AR-HMDs (Setup seen in 2.1).

2. Gesture only (producing a gesture set) and speech only elicitation study to highlight the individual strengths of these input modalities and a co-occurring gesture and speech elicitation study to highlight the synergies found when combining those modalities.

3. We present findings on the timing windows and syntax of co-occurring gesture and speech interactions and compare that with the syntax used in speech only interactions.

4. Design guidelines for AR interactions based on the synergies and individual strengths of gesture and speech interactions.

Multimodal Elicitation

In contrast to multimodal fusion designs, where input recognition and integration is often tested [57], we used participatory design guidelines [8] to work with the users to find which inter-

actions they would naturally want to use. This information can be used to help improve recognizer systems' accuracy and design user-centric interactions within AR-HMD building environments.

2.3.3 Why Gestures and Speech?

Interface design must be intuitive [58]. There is a large body of work on gesture and speech in human to human communication [53–55], and human computer communication [59–61]. An interface that mirrors human to human interactions could reduce the learning time needed for technology use. With that in mind, it is important to have systems with multimodal (e.g., gesture and speech combined) as well as unimodal (e.g., gesture or speech alone) interaction capabilities. Gestures and speech together constitute language [53]. They have bidirectional influence and obligatory influence on each other, which is to say that people typically consider both at the same time [54].

Using multimodal inputs has many benefits, particularly when dealing with gestures and speech combined. Gesturing when co-occurring with speech has been shown to help lower the cognitive load of a task [62], there are hints at sped up task completion time, and even lower error rates [61]. Each information stream (gesture, speech) contains non-redundant information [63] which can facilitate the disambiguation of the inputs from the other channel [64–66].

Given the option of using gestures, speech, or both combined participants used both 60% to 70% of the time [60, 67]. This can be exploited to help improve recognition accuracy [68]. Users feel that interactions are more natural when they have multiple input modalities and can choose the one that best suits them [69, 70]. The ability to have true multimodality could further improve their interactions.

Current AR-HMDs (i.e. Magic Leap One and Microsoft Hololens) are built with gesture sets that are limited and likely designed for recognition accuracy, not ease of use. For example, Magic Leap's "C" gesture is fairly easy to detect (being a static symbolic gesture) but may not be the most natural. Additional examples can be found in the other default gestures for the Magic Leap One and the Microsoft HoloLens 1. Occasionally gesture sets are derived from users; however, these

are often expert users [71]. People typically prefer user-defined gesture sets to expert-designed sets [8]. There is also evidence that elicited gestures are up to 24% more memorable [72].

The gap between traditional input devices and combined gesture and speech inputs is being minimized by advances in technology, soon gesture and speech inputs will be more efficient than traditional input devices [70].

Switching to these modalities is no trivial task. When using AR-HMDs, issues include gestures for ego-centric cameras such as the head mounted cameras on most HMDs, self-occlusion, device field of view (FOV), natural feeling interactions, common speech mappings, and timings of co-occurring gestures and speech when in virtual environments. This work tackles some of those issues and provides information on the individual and joint strengths of these modalities, a consensus gesture set, co-occurring gesture and speech timing information, and design guidelines to use when developing building applications for optical see-through augmented reality head-mounted displays.

2.3.4 Previous Work

Gesture elicitation is a study design that can help us map gestures to actions for emerging technologies. The elicited inputs have the goal of being highly discoverable to novice users of systems [8]. Elicitation studies also allow us to better understand user behavior. Elicitation studies have found that people use larger motions for larger objects when attempting the same action [73, 74], and that there is a preference for upper-body gestures even when a whole-body system is available [75]. Most commonly these studies have been conducted using Wobbrock et al.'s methods [8, 76], later refined by Vatavu and Wobbrock [49, 77] (variations exist [78]). This study used gesture elicitation, as well as mutlimodal gesture and speech elicitation, which is less common [43, 51].

Gesture Elicitation

These methods normally include the use of a Wizard of Oz (WoZ) experiment design. WoZ experiment design is a way to remove the gulf of execution between the participant and the sys-

tem [8]. In a WoZ elicitation experiment, a participant is shown a command to execute such as *move left*. This command is called a referent. Then the participant provides some sort of input proposal for that referent and behind the curtain, so to speak, an experimenter triggers the recognition of that input. In the experiment presented here that would look like a participant proposing a gesture (in the gesture modality) to move a virtual object left, then the experimenter, upon seeing this proposal, triggering the movement of the object. In this way, inputs can be designed for emerging technologies without perfect recognizers existing. After all the input proposals are collected they are binned into equivalence classes and measures of consensus between participants are used to generate input set proposals. This process is elaborated on later.

Many follow-up studies have created gesture sets using gesture elicitation [79, 80]. The popularity of gesture elicitation can be seen in the variety of the studies that use it, from multi-touch surfaces [80, 81], and mobile devices [82], to internet of things home sets ups [83]. Efforts to enhance further elicitation studies have led researchers to devise alternatives that extend beyond surface-computing devices, such as using multi-touch and mid-air devices in tandem [74, 84] and using multi-touch devices to control physical objects through virtual representations of said entities [85]. Imposing constraints on the users' motion has also led to new elicitation studies primarily concerned with defining and investigating gesture sets suitable for both impaired and non-impaired users [38, 86].

Gesture and Speech Studies

Gesture and speech input modalities have been studied for some time. Many studies have looked at ways of combining them as input channels using multimodal fusion modals [57, 59, 92, 93]. The goal of those studies was to implement recognition systems. Studies have also looked at the timing windows of co-occurring gestures and speech [9]. There is work on the usability of limited gesture sets [94] and constrained speech dictionaries [89]. Those types of works are aimed at understanding some combination of the feasibility of gesture and speech inputs, the adaptability of people to constrained inputs, and the implementation of fusion models for gesture and speech

Table 2.1: Previous gesture with speech elicitation studies

Authors	Display used	Consensus set made	Paired elicitation	Use case	Gestures accepted	Independent testing of modalities
Hauptmann et al. [87]	2d Screen	No	No	Graphic manipulation	Mid-air	Yes
Mignot et al. [88]	2d Screen	No	No	Control a process	Touch	No
Bourguet [89]	2d Screen	No	No	Explanations of process	Pointing	No
Carbini et al. [57]	2d Screen	No	Yes	Tell a story	Mid-air	No
Lee et al. [9]	2d Screen / handheld AR	No	No	Object manipulations	Mid-air	No
Morris [51]	2d Screen	Yes	Yes	Web browsing	Mid-air	No
Anastasiou et al. [90]	Room	No	No	Accessibility	Mid-air	No
Robbe [89]	2d Screen	No	No	Constrained speech dictionary	Touch miming and pointing	No
Khan et al. [43]	2d Screen	Yes	No	Computer aided design	Mid-air	Gesture / gesture or speech
Irawait et al. [91]	Optical see-through AR-HMD	No	No	Object manipulations	Open hand gestures	Gesture / gesture with speech
The study presented here	Optical see-through AR-HMD	Yes	No	Object manipulations	Mid-air	Yes

Legend: AR: Augmented Reality, HMD: Head mounted display, Miming gestures: charade like gestures

recognition. Those works typically start with defined acceptable inputs, maybe "open palm swiping" in the case of gestures [94], then test usage from there.

The work presented here is very different in that there are no constraints imposed on input proposals. Participants are free to generate any proposal that they feel is best suited to the

referent displayed. There have been previous studies on gesture and speech interactions. 2.1 shows a list of studies that use WoZ methods to observe or elicit gestures and speech interactions. Most of those studies did not have the goal of generating a consensus set of inputs. While a few of them did observe mid-air gestures [9, 43, 51, 57, 87, 90], some only looked at a subset of gesturing such as pointing gestures [89, 95], paddling gestures [91], or 2 dimensional (2D) gestures [88, 89]. The work presented here examines any gesture and / or utterance that a participant feels is appropriate for a given referent.

The study that is most similar to this is a gesture and speech elicitation study done for developing commands for a television-based web browser [51]. Participants were placed in paired elicitation sessions where the dyads of participants made proposals together. The referents were read out loud to the participant. For the referent *move left* the experimenter would read "move left". Participant dyads were given the choice of using either gesture, speech, or both; however, the modalities were not tested individually. Commands for web browsing on a television (i.e. "refresh page") are decidedly different from the commands needed to manipulate objects in optical see-through AR environments.

A second similar study did gesture and speech elicitation for computer-aided design (CAD) programs to be used with 2D screens [43]. This experiment tested gesture alone, then gesture with speech. They provide a consensus set of utterances and gestures. This study chose to show the referents' action in the form of an animation as opposed to as text. For the referent *move left* the participant would see the virtual object moving left. This study is domain-specific to CAD program usage. Previous work has found that prompting users to gesture with 2D screens compared to 3 dimensional (3D) objects can impact the production of gestures [96]. Additionally, Khan et al. informed users that they were describing referents to another person though use of a video system. The notion of describing a referent to a person compared to executing a referent in a system is an important distinction. This work also extends the work of Khan et al. by providing the timing information of co-occurring gesture and speech interactions.

All of the studies shown in 2.1 have furthered the field of gesture and speech interactions. Still, those studies are different from the work presented here in some major ways. Including the pairing of participants, domains of application, and how the referents are presented. Most of those studies only tested interactions in a single pass where participants proposed speech alone, gesture alone, or both together. Whereas this paper tests each modality independently. The last row of 2.1 shows the methods used in this study, to be compared with the other works. This study will help to further gesture and speech elicitation methods, AR-HMD interactions, object manipulations in 3D space, and finding differences between when speech alone is used and when co-occurring gestures and speech are used.

While the research presented here is not on gesture recognition or multimodal input fusion, elicitation can provide important findings for future recognizers (including findings from this research). Recognition of gestures has been attempted in many ways; however, it has not often been done with AR-HMD's and ego-centric cameras.

Elicitation Criticisms

Elicitation methods have received criticism in two major areas. First, it was suggested that common consensus metrics were too permissive because they do not account for the base chance of randomly selecting a proposal for a given referent [50]. Tsandalis proposed using Fleiss' kappa and a chance agreement term in addition to those metrics to address this [50]. We have analyzed our data using those statistics to alleviate this concern. Second, there is a concern that given the exposure to existing devices or gestures, elicitation may be biased (i.e., legacy bias). This has been examined, and various ways to incorporate it [75, 97] or reduce it [38, 98] have been introduced. However, other than priming [42], no reduction methodology has shown promise, except for physical constraints [94], but constraints are infeasible in some cases. Some work has shown that legacy bias can be beneficial in finding gestures for abstract tasks [99].

2.3.5 Methods

This work performed an elicitation study using the WoZ methods to find natural feeling gesture, speech, and gesture with speech interactions for the manipulation of rendered 3D objects in optical see-through AR environments. The input modalities used were Gestures (G), Speech (S), and Gesture with Speech (GS). Each modality was tested independently in a within-subjects experiment design. Our methodology is derived from our previous work and the literature already described. These include agreement rate ($\mathcal{AR}$)[4], co-agreement rate ($\mathcal{CR}$), and the V_{rd} significance test [8, 49, 77]. When reporting overall agreement rates for gesture proposals, we also make use of Fliess's Kappa coefficient (κ_F) and the chance agreement term (p_e) as described by Tsandilas [50].

Both speech and gesture proposals were annotated based on the video data from the exo-centric and ego-centric cameras. Proposals then were binned into equivalence classes by the experimenter. Gestures were binned based on the direction of movement, and hand pose. Hand poses were "grasping" where all fingers were closed, "pinching" where just the thumb and index or thumb index and middle fingers were touching, "open" where all fingers were extended, and "index finger" where only the index finger was extended. Previous work showed that users care less about the count of fingers used than the hand pose used [84]. Movements were based on the axis of movement. For example, translations right and left were both considered movements on the y-axis. If a gesture could not be binned in this manner it was given its own class(i.e. tracing a square). For speech calculations, words were binned only if they were nearly identical. Saying "move forwards" and "move forward" were considered the same where "move towards" would be different.

The original metric for consensus is the agreement index which involves the proportion of participants proposing equivalent gestures [76]. This metric was changed to $\mathcal{AR}$ which addresses some of the issues with the original formula, adjusting the output values to between 0 and 1 [49]. $\mathcal{CR}$ is defined as a measure of shared agreement between two referents. It is calculated as the count of pairs of participants that are in agreement for two referents over the total possible pairs of

[4]Please note that agreement rate $\mathcal{AR}$ uses a different font to avoid confusion with AR for augmented reality.

participants [49]. For speech alone the consensus-distinct ratio (CDR) was used. The CDR is the percent of equivalent proposals given by more than two participants for a given referent [51].

Participants

The study consisted of 24 volunteers (4 female, 20 male). Participants were recruited using emails and through word of mouth. Ages ranged from 18 - 43 years (Mean = 23.32, SD = 5.23). All participants reported heavy computer usage but limited video game usage. Two participants were left-handed. Eleven participants reported less than 30 minutes of Microsoft Hololens 1 usage before this experiment. Seven participants learned English as a second language and reported fluency in English.

Apparatus

This experiment was conducted using a Magic Leap One optical see-through AR-HMD. The WoZ system was developed in Unreal Engine 4.23.0. A Windows 10 professional computer with an Intel i9-9900k 3.6GHz processor and an Nvidia RTX 2080Ti graphics card was used for development. Data was recorded on the Magic Leap One. In addition, we used a GoPro hero 7 black (to record an ego-centric view of the interactions) and a 4k camera (to record an exo-centric view of the interactions). Each referent was shown 50 centimeters in front of the user. Users were given an on-screen aid to tell if their hand/hands were inside of the FOV of the device. This aid showed one hand on each side of the screen in red unless a hand was seen. If a hand was sensed the corresponding aid (left to left, right to right) would turn white, as shown in 2.2.

Referents

Table 2.2: Referents used by category

Translation	Rotation	Abstract	Scale
Move (Left / Right)	Roll (C / CC)	Create	Enlarge
Move (Up / Down)	Yaw (Left / Right)	Destroy	Shrink
Move (Towards / Away) from self	Pitch (Up / Down)	Select	

Legend: C: Clockwise; CC: Counter Clockwise

Figure 2.2: Left:participant view in experiment, Right: participant

Referents (i.e. actions) for canonical manipulations [46] including selection, scaling, translation (on x,y, and z axes), and rotation (about x,y, and z axes) were used. In addition, application-specific manipulations [46] which included create and delete were used. All the referents are listed in 2.2. The goal of this study is to create an interaction set for object manipulations in any sort of virtual environment that uses building tasks (e.g., Lego-like applications). Specifically, when AR-HMDs, egocentric viewing, and multimodal inputs are used. Object selection was tested independently in the *select* referent. For the other referents, participants were told that they could assume the object was already selected. Referents were displayed as text. This decision was informed by previous work [43,51] and the results of pilot studies which are discussed further in the *Results* section.

Procedure

At the start of each session, participants completed an informed consent and demographics questionnaire. The questionnaire included questions about prior device usage, game usage, and handedness. Participants were then shown a 2-minute video with the instructions for the experiment. They were informed that they would be asked to complete a series of object manipulations using different modalities of input and within each modality (G, S, or GS) they could use whatever input they wished. For example, if the modality was gesture than any gesture proposed was accepted. Participants were then given a practice trial for each of the modalities. During this time, they were invited to ask questions, adjust the device, and play with the device's gesture sensing

range using the on-screen hand detection aid, see the left side of 2.2. Note that the gesture sensing was to add realism to the experiment but this experiment was a WoZ elicitation experiment.

Participants were presented the interaction modalities based on a Latin square division of blocks. For example, participants may have seen speech first, then after completing all the referents for speech, see the next modality (G or GS in this example). Referents were shown in random order. The object to be manipulated was a cube rendered approximately 50cm in front of the user. A cube was chosen to allow visual ques of rotations which would be more difficult to see with either a sphere or cylinder. A cube represents a basic object that most users have interacted with in the real world. Using a cube limited some of the object specific grips that could appear in interactions with complex shapes (hand pose matching uneven object surface).

The referent was shown as a text banner and above that, the interaction modality requested was shown. On either side of the cube was a hand that was either red with a line crossing it or white (left side of 2.2). The hands indicated whether or not a participant's hand was in the camera's field of view. An example of a referent and corresponding gesture proposal is shown in 2.2. After a proposal was made by the participant the virtual object would execute that referent and the next referent would be loaded. In the G and S blocks this execution occurred when any proposal was given. To ensure constancy of proposal modality in the GS block both an utterance and a gesture had to be proposed before the referent was executed. After each interaction modality, the NASA TLX [35] survey was administered.

2.3.6 Results

The agreement rate ($\mathcal{AR}$), co-agreement rate ($\mathcal{CR}$), and (V_{rd}) statistic were used to quantify consensus among participants. Fliess's Kappa coefficient (κ_F) and the associated chance agreement term (p_e) [50] were used when reporting the overall agreement rates for the gesture proposals. Where applicable, the appropriate statistics were computed using the AGATe 2.0 tool (AGreement Analysis Toolkit) [5]. For the speech proposals, the consensus-distinct ratio (CDR) was used [51].

[5] Available at http://depts.washington.edu/acelab/proj/dollar/agate.html

The agreement rate $\mathcal{AR}$ is defined as the number of pairs of participants in agreement with each other divided by the total number of pairs of participants that could be in agreement. Shown formally for a single referent r in 2.3, where P is the set of all proposals for referent r, and P_i are the subsets of equivalent proposals from P.

$$\mathcal{AR}_r = \frac{\sum_{P_i \subseteq P} \frac{1}{2}|P_i|(|P_i| - 1)}{\frac{1}{2}|P|(|P| - 1)} \tag{2.3}$$

Pilot Studies

Two versions of this elicitation experiment were run on pilot groups consisting of 6 people each. In one, we displayed the referents as text (2.2), in the other we showed the action of the referent then asked for proposals. As an example, if the referent was *move left*, in the first set up the screen read "move left" and participants were asked to propose a command to execute that referent (similar to [8, 51]). Upon generation of that proposal, the virtual object would move. In the second design, the virtual object would move then participants were asked to generate an appropriate command proposal (similar to [43]).

During the speech block of the pilot study where referents were displayed as text participants would commonly repeat the referent displayed. If the referent was *move left* the utterance was also "move left". This is not entirely unreasonable. In the pilot study without text, for simple translations, the most frequent utterances were "move" and the direction such as "left". This repeating of referents, either the entire referent or a sub-portion of it can also be seen in the results of Morris [51]. An example from that study is that when given the referent *open new tab* the top utterances were "new tab" and "open new tab".

In the version with referents shown as movement, people would nearly always propose a gesture that was as close as to a one to one manipulation with the object's motion as was possible. For rotations, people would twist their wrist into uncomfortable positions to try and match the object's motion. For the abstract referents, people's gestures would mirror whatever animation was shown. If the virtual object was materializing from right to left, their hand moved from right to left. More

troublingly, none of the participants understood what was being asked of them when the referent was *create* and the virtual object appeared with no animation. The effect referent animations biasing gesture production can be seen in [43]. Examples from that study include the proposed gestures for the *orbit* and *pan* referents which have participants' top choice gestures mirroring the visual motion of those referents.

Due to the evidence of priming gestures found when showing the referent as an animation, we have chosen to display referents as text. This set up can be seen in 2.2. There is no perfect solution for how this experiment should be run. The text banners had less priming on the gesture alone and the gesture and speech conditions. In the case of speech alone, some speech was primed to repeat the referent as displayed, also seen in [51]. This was not always the case. For some referents, such as the rotational referents, the utterances "tilt", "rotate", and "spin'' occurred with high frequency. These utterances were also found in the pilot study where users were shown the animation of the referent with no text. We believe that while the individual utterances found in the speech block should be observed skeptically, the overall results still yield insights into what utterance people will gravitate towards using in augmented reality manipulation tasks, as Morris's work with similar biasing yielded insights into appropriate speech commands for web-browsing on large screen displays [51]. Additionally, because speech was examined alone, differences in what utterances occur alone versus what utterances occur when accompanied by gestures can be examined.

Gesture Only Block

The overall agreement rate observed for the Gesture block was .353 with $\kappa_F = .317$. The low chance agreement term ($p_e = .058$) used in Fleiss's Kappa coefficient indicates an agreement beyond chance [50], allowing us to consider rates above 0.3 as high levels of consensus between participants given our N of 24 based on the simulations of varying agreement distributions found in [49]. The agreement rates for each referent are given in 2.3 and shown as numbers in 2.6 and 2.5.

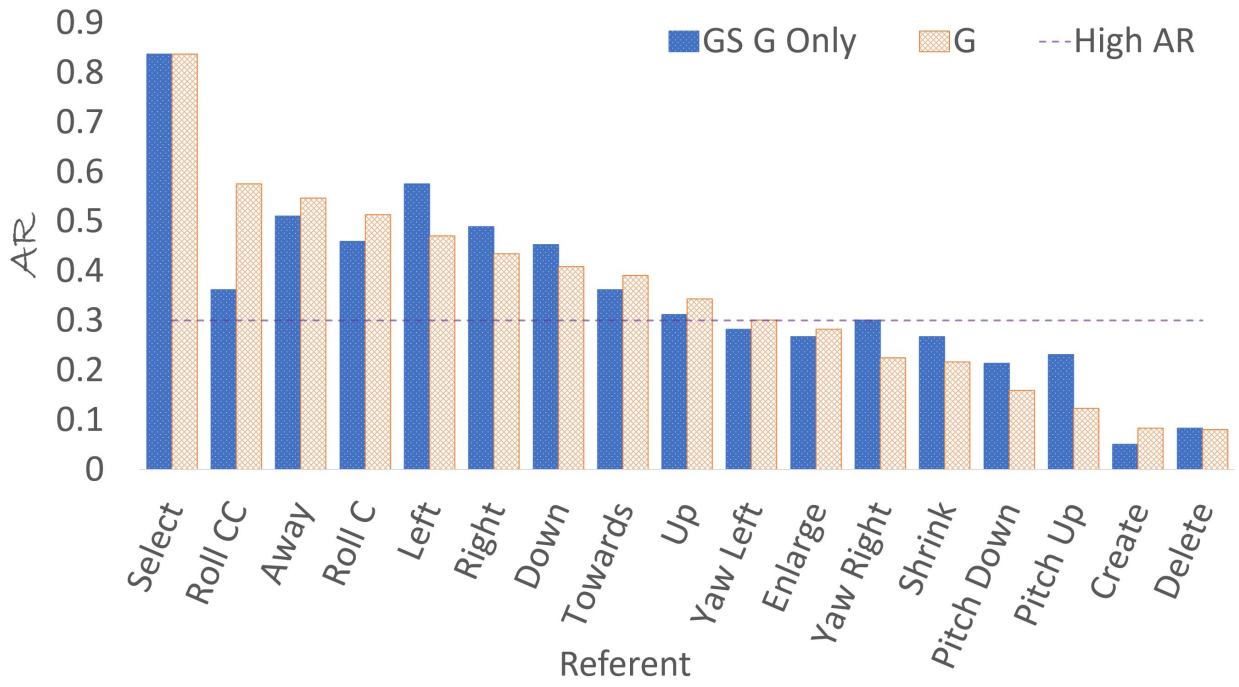


Figure 2.3: Agreement rates for gestures in the gestures block (G) and the gesture with speech block (GS); C: Clockwise; CC: Counter Clockwise

The effect of referent type on agreement rates was observed to be significant ($V_{rd(16,N=408)} = 856.872, p < .001$). The highest single agreement rate belonged to the referent *Select* ($\mathcal{AR}_{Select} = .837$), which may be due to the legacy bias from the smart phone (e.g., iPhone). The more abstract referents, *Create* and *Delete*, exhibited extremely low agreement rates ($\mathcal{AR}_{Create} = .083$, $\mathcal{AR}_{Delete} = .08$).

The referents involving a physical translation (*up, down, left, right, away*, and *towards*) had high gesture agreement among participants (average $\mathcal{AR} = .433$). Among these translational referents, the direction of motion displayed a significant effect on agreement rates ($V_{rd(5,N=144)} = 41.446, p < .001$), with *away* achieving the highest individual agreement ($\mathcal{AR}_{Away} = .547$). While no significant difference in agreement was found between *right* and *left* ($V_{rd(1,N=48)} = 2.174, p = 1$), a significant disparity was observed for referents *towards* and *away* ($V_{rd(1,N=48)} = 18.677, p < .001$).

For the three pairs of translational referents, *Right* and *Left* had the highest co-agreement rate ($\mathcal{CR}_{Right,Left} = .37$), indicating that 37% of all participant pairs agreed on both referents.

While the average of the rotational referents (average $\mathcal{AR} = .316$) was comparable to the translational group, this was primarily due to the out sized contribution of *roll clockwise* and *roll counter clockwise*. Presumably, this high agreement for *roll* (average $\mathcal{AR}$=.545) can be attributed to the implied clock metaphor with participants pantomiming the rotation of clock hands. Among the rotational referents, the impact of referent type on agreement is considerable ($V_{rd(5,N=144)} = 271.232, p < .001$), reflecting the great disparity between *roll*'s elevated agreement and the relatively low consensus observed for *pitch* ($\mathcal{AR}_{PitchUp} = .123$, $\mathcal{AR}_{PitchDown} = .159$). Moreover, for the three pairs of rotational referents, 39% of all pairs of participants agreed on both *Roll Clockwise* and *Roll Counter Clockwise* ($\mathcal{CR}_{CW,CCW} = .391$).

It should be noted that although *Shrink* and *Enlarge* exhibited comparable agreement rates ($\mathcal{AR}_{Enlarge} = .283$, $\mathcal{AR}_{Shrink} = .217$), there was little agreement among pairs of participants for both referents ($\mathcal{CR}_{Enlarge,Shrink} = .123$).

Speech Only Block

Table 2.3: Consensus-distinct ratio for the speech and gesture with speech blocks by referent type

Category of referent	Gesture and Speech	Speech
Abstract	39.52%	24.52%
Rotation	44.72%	39.76%
Scale	32.50%	39.29%
Translation	53.89%	61.11%

Displaying the referent in elicitation studies [37] and reading the referent out loud in gesture and speech elicitation studies [51] both have precedence. As previously noted, these practices can prime the utterances proposed. Often the referent as displayed was repeated, however, this was not always the case. When it was, the referents were simple such as “move left”. The repetition could be in part due to priming, though it could also be that there are few aliases for the phrase “move left”.

The average CDR for each category of referent (2.2) is shown in 2.3. The translations hold the highest CDR. This can be interpreted as the translation referents having the least disagreement on the appropriate utterance proposal. Translations were nearly always the direction of movement alone (i.e. "left", "up") or a <action> <direction> pair (i.e. "move up"). The scale and rotational referents had more disagreement shown by the lower CDRs at 39.29% and 39.76% respectively. The lower CDR for scaling referents was due to a high number of aliases for each proposal, in the case of *expand* they included "grow", "zoom", and "expand". For rotations the phrases "rotate", "spin" and "tilt" paired with a direction such as "up" were proposed. "Spin" and "rotate" were commonly used for *Yaw*, "tilt" for *pitch*, and "roll" for *Roll*. "Select" was proposed by each participant for *Select*, however, there was disagreement on how to indicate the virtual object. Participants commonly said "select cube" but some said "object", or "that". The referent category with the lowest CDR was abstract referents at 24.52%. These being *create* and *destroy*. This is interpreted as meaning for the abstract tasks there was high disagreement between proposals. Commonly proposals used the word "create" or 'destroy" but disagreed on the object identifier, as seen with *select*.

We believe that aliasing commands would be beneficial when dealing with unimodal speech, as do [8,51]. While our participants were told that they could use any utterance that they wanted, they primarily stuck to <action> <direction> or <action> <object> <direction> phrase structure. The rates for the syntax are found in 2.4. A chi-square test of independence showed that there was a significant association between block and syntax choice $X^2(2, N = 408) = 71.28, p < 0.001$. For most commands, the direction and type of manipulation were proposed (e.g., "move left", "roll right"). For commands with lower CDR we recommend aliasing some of the manipulation terms. Specifically, "spin", and "roll" were used interchangeably. For decreasing object size the combination of "smaller", "small", and "shrink" would cover 75% of proposals.

Multimodal Block: Gesture and Speech Combined

This section provides three analyses of the co-occurring gesture and speech block (i.e. multimodal interactions). First, the gesture portion of this block was isolated for comparison with the

Table 2.4: Usage of syntax format by block

	Other	<action> <direction>	<action> <object> <direction>
GS	24.31%	62.75%	16.91%
S	11.52%	86.27%	2.21%

Legend: S: Speech block; GS: Gesture and Speech block; other: single or many word command

gesture only block (2.3.6). Second, the speech portion of this block was isolated for comparison with the speech alone block (2.3.6). Third, the gestures and speech from this block were analyzed. This breaking apart of the analyses allows for a more thorough examination of the data and better comparisons with the other modalities (previously described in 2.3.6 and 2.3.6).

Gesture in GS Block This is the analysis of the gesture proposals alone from the GS block. The overall agreement score observed for the gestures in the GS block was .357 with $\kappa_F = .318$. The chance agreement term in Fleiss's Kappa coefficient ($p_e = .057$) indicated an agreement beyond chance [50], allowing us to consider agreement scores above 0.3 to be meaningful [49]. The agreement scores for each referent of the GS block are displayed in 2.3.

The influence of the type of referent on the agreement rates was, again, measured to be statistically significant ($V_{rd(1,N=48)} = 770.497, p < .001$). As in the Gesture block, *Select* had the highest individual agreement rate ($\mathcal{AR}_{Select} = .837$), while again *create* and *destroy* ($\mathcal{AR}_{Create} = .051$ and $\mathcal{AR}_{Delete} = .083$) could, at best, be described as negligible agreement.

The translational referents maintained a high gesture consensus (average $\mathcal{AR}$= .451) over the GS block and the agreement rates were, again, significantly influenced by direction ($V_{rd(5,N=144)} = 87.488, p < .001$). While the referent *Left* had the highest single agreement rate ($\mathcal{AR}_{Left} = .576$), the referent *away* still retained a high consensus with $\mathcal{AR}_{Away} = .511$. The dichotomous pair (*Up,Down*) showed the largest variation in agreement with $V_{rd(1,N=48)} = 32.362, p < .001$. Not surprisingly, the pair (*Right,Left*), with a high co-agreement rate of $\mathcal{CR}_{Right,Left} = .402$, only displayed a slightly significant difference in agreement rate ($V_{rd(1,N=48)} = 8, p < .01$).

Overall agreement for the rotational referents (average $\mathcal{AR} = .309$) was lower than the translational group and the impact of referent type on consensus, while still present, was decidedly

diminished ($V_{rd(5,N=48)} = 77.996, p < .001$) as compared to the gesture block. Ostensibly this can be attributed to the decreased difference between *roll*'s high agreement (average $\mathcal{AR} = .411$) and *Pitch*'s relatively low agreement (average $\mathcal{AR} = .223$).

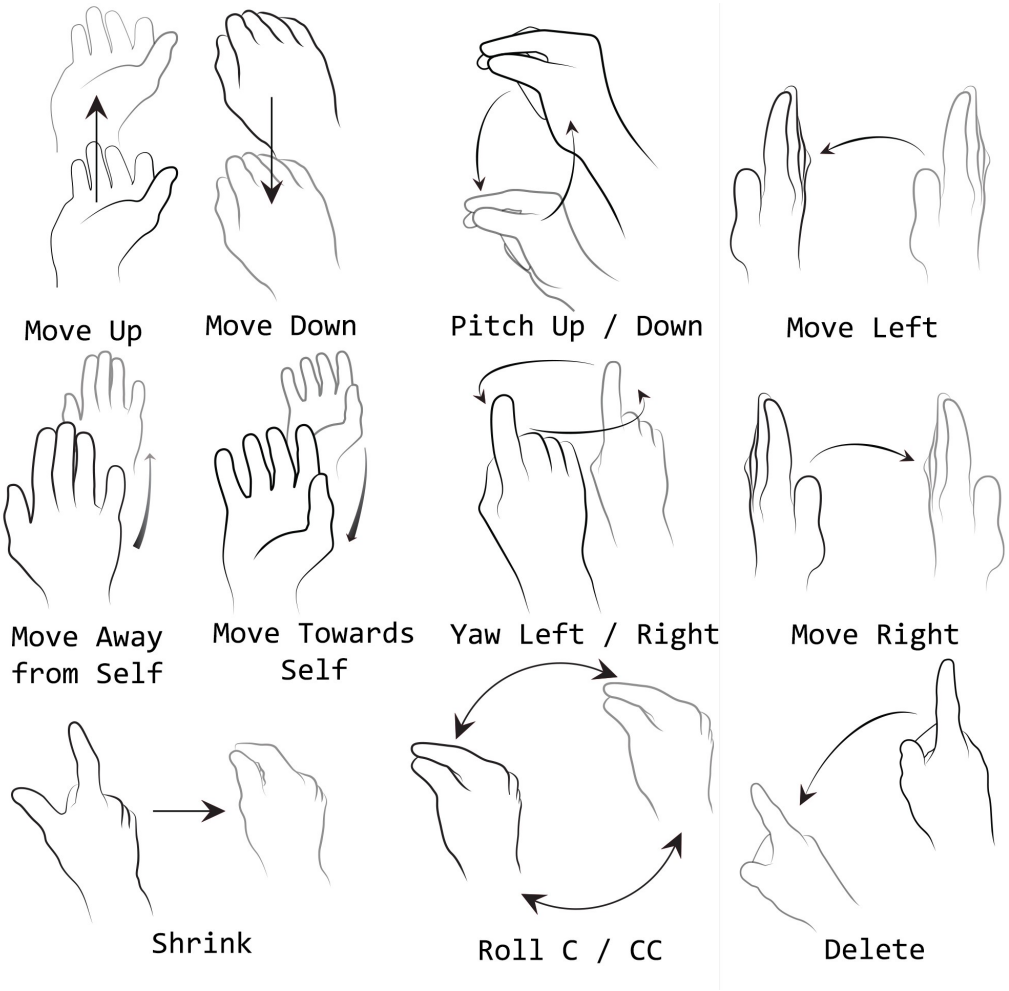


Figure 2.4: Proposed gesture set; C: Clockwise; CC: Counter Clockwise; Bi-directional gestures indicated with double arrows

Speech in GS Block The speech alone in the GS block achieved higher CDR for most categories of referent indicating more agreement in proposals for a given referent (2.3). This was due to less disagreement on the direction and object identifiers. Due to the pairing of gestures with speech participants would indicate the direction of movement with their finger (a finger tracing a circle in the case of yaw, see 2.4). When using this style of command, participants would point to the

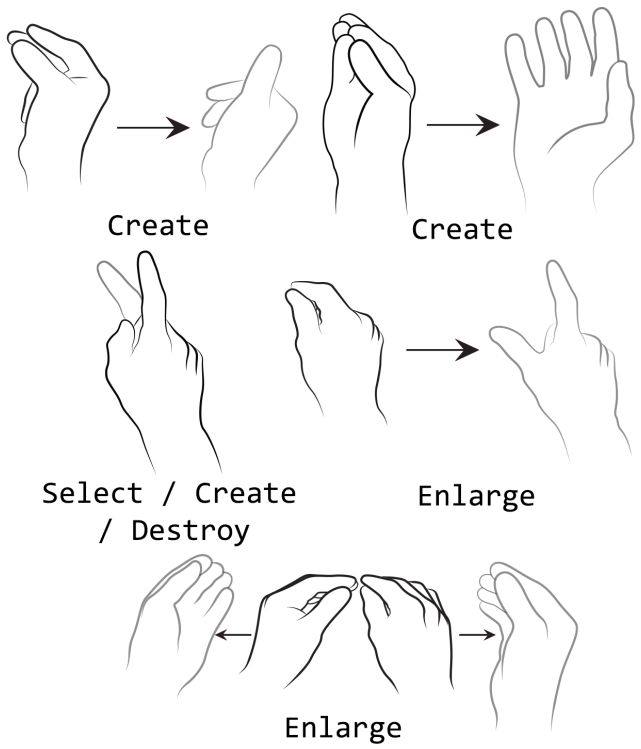


Figure 2.5: Gestures with ties

Table 2.5: Tied Gestures

Input	Referent	Gesture	$\mathcal{AR}$
GS	Create	Bloom	0.05
GS	Create	Legacy tap	0.05
GS	Delete	Legacy tap	0.08
G	Enlarge	Two Hand Grow	0.28
G	Pitch Up	Push up	0.12
GS	Pitch Up	Circle forward grabbing	0.23

Legend: C: Clockwise; CC: Counter Clockwise, G: Gesture block; GS: Gesture and Speech block

object first then initiate their command. Translations CDR dropped to 53.89%. In the speech alone block participants would most commonly (86.27%) use the "move" and direction phrases together (2.4). When gestures were also allowed participants would default to only using the direction phrase and a gesture or less commonly "move" alone and a gesture indicating the action ("Other" column in 2.4). Seen as proposing "right" and a pointing gesture. During interviews held after the experiment, most participants indicated wanting to do translations via gesture manipulations

Table 2.6: Winning Gestures

Referent	Gesture	$\mathcal{AR}$ G	$\mathcal{AR}$ G in GS
Away	Push away	0.55	0.51
Delete	Swipe R to L	0.08	0.08
Down	Swipe down	0.41	0.45
Enlarge	Legacy zoom in	0.28	0.27
Left	Swipe left	0.47	0.58
Pitch Down	Circle forward grabbing	0.16	0.21
Right	Swipe right	0.44	0.49
Roll C	Circle C grabbing	0.51	0.46
Roll CC	Circle CC grabbing	0.58	0.36
Select	Legacy tap	0.84	0.84
Shrink	Legacy zoom out	0.22	0.27
Towards	Pull towards	0.39	0.36
Up	Push up	0.34	0.31
Yaw Left	Circle pointing up	0.3	0.28
Yaw Right	Circle pointing up	0.23	0.3

Legend: C: Clockwise; CC: Counter Clockwise; G: Gesture block; GS: Gesture and Speech block

only. The same pattern is seen in the scaling referents. Participants had more disagreement with the speech to use in this condition.

What is important to note here is that the speech commands for abstract referents had less disagreement in the gesture and speech block. This indicates that while gesture alone is well suited to translations gesture with speech is more suited for abstract commands.

Speech with Gestures in GS Block With the vastly larger proposal space offered when giving an utterance with a gesture the $\mathcal{AR}$ metric breaks down. This is to say that while mid-air gestures are somewhat limited in the number of proposals available, speech is far more nuanced. The combined pairings of gesture with speech are too varied for the use of $\mathcal{AR}$ without artificially binning words into equivalence classes. When observing the pairing of gesture and speech as a whole we find that 10.42% of the participants using the <action> <direction> pattern in speech used a <action> <gesture> proposal in gesture with speech. For translation referents, this looks like a participant

saying “move” and swiping with their finger in a direction. With rotations, participants would say “rotate” or “spin” and tracing a circle with their finger (2.4).

Timing of co-occurring gestures and speech The times between when a gesture was initiated and an utterance was initiated in milliseconds were (M = 151.31, SD = 120.24, Median = 130). These were measured by the time of any hand starting to move to the first sound emitted, or utterance to gesture if the utterance occurred first. This data took a non-normal distribution (Shaprio-Wilks $p < 0.001$). We speculate that this is because on several occasions participants had to stop to think about which rotation they were performing, heavily skewing the time and causing many outliers. Based on a Wilcoxon Signed rank test ($p < 0.001$) we can assume that the true median for the data is different from zero. In this study the gestures nearly always started before speech.

This result is similar to previous results [9, 95]. The results found in this study are primarily manipulative gestures whereas the results in previous work were experimenter defined deictic gestures (i.e. pointing gestures) [95] and spontaneous gestures that were primarily deictic [9]. This shows that the commonly found timing window for co-occurring gestures and speech exists for both deictic and manipulative gestures. This result also shows that gesture and speech interactions in AR-HMDS have similar timings [100] and patterns of occurrence [101] as ones outside of them.

NASA Task Load Index

The NASA TLX is a survey that is used to measure a participant’s perceived workload for a given task [35]. The mean scores for the NASA TLX overall workload for the three blocks are shown in 2.7. An ANOVA showed that there is evidence of a difference between the means of the three groups (df=2,69, p = .053). We take this to mean that producing both gestures and speech combined had a higher perceived workload than producing either individually. This follows the logical intuition that producing two inputs is harder than producing one. We speculate that given an interaction set, thus not needing to create proposals, there would be lower perceived workload with multimodal inputs. As is seen in other multimodal studies [67, 102]. Admittedly a p-value

of 0.053 is not equal to 0.05. That said with previous findings suggesting the same conclusion we speculate that given a larger N a difference in the overall workload would have been found.

Table 2.7: Average NASA TLX scores by block

	Gesture	Speech	Gesture and Speech
Mean	39.3	33.5	43.5
SD	13.4	15.6	13.3

Trial Times

The times for each trial as measured by when a referent was presented and the participant started a gesture or utterance in milliseconds are shown in 2.8. Linear contrasts showed that there is a significant difference between both gestures and speech versus gestures with speech (both $p < 0.01$, df = 1216). There was no significant difference between gesture alone and speech alone trial times (P = 0.91, df = 1216). Which follows what is expected; producing gestures and speech took longer on average than just producing gestures or speech alone. As this was measured from when either a gesture or utterance was initiated this implies that the gestures and speech block took more planning before a response.

Table 2.8: Average trial times by block in ms

	Gesture	Speech	Gesture and Speech
Mean	282	287	323
SD	158	158	186

Consensus Set

Most referents had a single most common gesture, seen in 2.4. Some referents had ties shown in 2.5. The ties for *create* predominately occurred in the GS block. All of the manipulative gestures were symmetric and bi-directional. Meaning that *roll clockwise* would be tracing a clockwise circle

and *roll counterclockwise* would be tracing a counter-clockwise circle in the same manner. In the G block people swiped down and to the left for *delete* as seen in 2.4. When speech was allowed some people switched to the taping gesture (*Select* / *Create* / *Delete* in 2.4) and using a word for the action. A tie was found between the *enlarge* proposals where both the single hand legacy zoom in gesture and a two-handed expansion gesture were produced (2.5). The expansion gesture is the only two-handed gesture that occurred with enough frequency to be shown. There were a number of two-handed gestures proposed for translation that were symmetric bi-manual versions of the single-handed gesture (two hands pushing forward).

2.3.7 Discussion

In contrast to the findings of Khan et al. [43], this study found that most gesture proposals were one-handed. There were differences in the gestures produced for scaling which were predominately bi-manual hand expansions in Khan et al. and a mix of bi-manual expansions and the legacy touchscreen zoom in zoom out gesture in this work(2.5). The translation gestures found in this study were nearly always direct manipulation gestures. Khan et al. found bi-manual direct manipulations and bi-manual path tracing gestures for translations. Rotations were comparable between the two studies. For rotational referents the "hold and rotate" gesture found by Khan et al. was similar to the pinching roll here (2.4). Speech found by Khan et al. was similar to the speech found in our study for the translations where "move" was the most common choice in both studies. It is difficult to compare results for the other referents as either the axis of movement is not listed or the referents do not match.

The differences found in gestures produced between these studies could stem from the participant believing they were interacting with a human versus a system. Another cause could be the way the referents were presented to participants. Interactions with a 2D screen may be formed differently than those in 3D space [96].

When comparing to the augmented reality gesture elicitation study done by Piumsomboon et al. the translation gestures for both studies were often open handed [99]. Rotations were varied

from previous work. Most rotations found here involved a pinch or index finger extended with movement following the path of a circle. Piumsomboon et al., encountered lose griped gesturing where a participant would grab the virtual object and rotate their wrist while holding it. The scaling gestures proposed in this study were commonly single handed (2.4) where the proposals found in Piumsomboon et al. were more often bi-manual [99]. The exception being the bi-manual "enlarge" gesture found here (2.5) which mirrored the uniform scale on the X-axis proposal [99]. Across both studies, most of the gestures found were reversible [99]. This is shown in the rotation and translation gestures in 2.4.

Scaling was comparable across these two studies presumably due to participants' legacy bias from interactions with multi-touch devices(e.g. cellphones). When differences were found it could be due to the difference in the presentation of the referents. Piumsomboon et al. showed referents as animations of the intended action where this work showed referents as text.

Individual Strengths

During the practice block, participants were encouraged to move both hands in front of the device sensors to see the range of the device's hand recognition, then instructed to use one or both hands as they deemed appropriate. Even so, participants tended to use one-handed gestures (2.4, 2.5). This mirrors what was found on multi-touch surfaces in [103–105] and mid-air full-body studies [37]. People tend prefer simple interactions over more complex ones [105]. We believe that the high number on one-handed interactions found in this study was due to the referents low level of complexity and that preference for simple interactions when possible.

Translation gestures shared high agreement rates for both the gesture and the gesture and speech blocks. Most often, participants reached forward to where the object was rendered and preformed a direct manipulation (2.4). For example, they reached out and pushed against the side of the cube to move it in any direction. Thirty seven percent of participant pairs agreed on the referents *right* and *left*. We interpret this as meaning that when manipulating virtual objects, using direct manipulation techniques for translations is more natural.

However, when dealing with rotations, we saw more indirect manipulations in the form of circles made in the air around the axis of the intended rotation (2.4). A few people reached out and rotated the object directly (most common for roll, some occurrences in yaw). It is also of note that on a few occasions in the speech only block participants would make tracing gestures with their finger (2.4) for rotational referents. We speculate that this was to help lessen the cognitive challenge of figuring out which rotation was necessary by transferring the mental process to their visuospatial sketchpad. This follows previous findings that gestures help lighten the cognitive load of speech-based tasks [62].

During interviews after the experiment, 18/24 participants said that gestures were preferred for translations saying that gesturing took less thought. As seen in 2.4 the most agreed upon proposals used reversible gestures for pairs of actions. This mirrors previous elicitation studies work [8, 99].

Select had the highest $\mathcal{AR}$ overall. This was due to the high occurrence of the legacy tap gesture (2.4). Legacy gestures were also produced for *Enlarge* and *Shrink*. Those being the two-finger zoom in /out from consumer touch screen phones. These gestures had a 12% co agreement rate. Meaning that while the gestures were highly agreed upon, pairs of participants were unlikely to agree on the same gestures for both referents. Legacy gestures are gestures that were used as inputs for previous technologies [98]. Legacy bias is viewed as negative when it does not utilize options available in the new input environment. This bias could be beneficial [37,94,97]. When appropriate for the new environment, legacy gestures provide the benefit of being more discoverable and more memorable to novice users [98].

Gesture and Speech Synergies

Delete and *create* had the lowest consensus in both the gestures and gesture and speech blocks. For these, speech might be the optimal input or a gesture derived by designers after doing a preference study. In the gesture and speech block, these referents had a higher CDR. Indicating that there was less disagreement between participants in the utterances proposed. Post-hoc analysis showed that while participants had less disagreement on the utterances proposed, they had higher disagreement on the appropriate gesture. Pointing to a location and saying “delete” or “remove” occurred

with some frequency but the rate of snapping and blooming gestures lowered the overall AR (2.5). Even so, we believe that the benefit of improving the CDR makes these abstract commands well suited for co-occurring gesture and speech inputs. Other work has shown that producing gestures for abstract referents is difficult for some users, further bolstering this argument [99].

When only speech was allowed, most people use <action> <direction> or <action> <object> <direction> syntax such as "move left" or "move the cube left". When switching to multimodal inputs, people used more deictic gestures paired with an <action> <phrase> or action-gesture paired with a manipulation phrase. This is seen as a pointing gesture and saying "move" followed by a finger flicking in the direction of the intended movement.

Participants would use gestures to help with speech in the speech only block indicating a preference for multimodal interactions for rotational referents. Disfluent language (saying "left" when you mean "right") can be reduced by up to 50% when using multimodal gesture and speech [106]. This is due to the difficulties that most people have with spatial information, which in this study were the difficulties found when determining the correct direction for the rotations. The gesture portion of these commands was typically a finger trace indicating the orientation of the rotation, which helps resolve the issue of finding the right language to execute the rotation. Five participants spontaneously gave degrees when presented with rotational referents. This added fine-grained turn control is a another compelling reason for enabling multimodal interactions for precise rotations.

An important finding of this study is that the median time between when a gesture starts and an utterance starts is 130 milliseconds. This finding can help researchers set up recognition windows for interactive multi-modal systems by indicating what lengths of time to wait between those input modes. Additionally, this finding helps bridge the work of linguistics [53–55] to human computer interaction. This shows that some of what is known about human to human communication extends into multi-modal interactions within AR environments. Similar findings have been seen for deictic gestures [9, 66, 91]. These findings presented here indicate that the timing windows for more generic manipulation gestures also conform to this pattern.

2.3.8 Design Guidelines

An optimal system would allow for unimodal gesture, unimodal speech, and multimodal gesture and speech interactions. While a large portion of users enjoy gesture and speech interactions [56, 67], some users still prefer unimodal interactions. For many things, direct manipulation should be available, particularly in the case of translations. For rotations, multimodal gesture and speech interactions should also be allowed. For every manipulation action, reversible interactions should be used. These could look like the gestures shown in 2.4. With speech, this is more difficult but possible in some cases where a word has a clear opposite (i.e. "create" and "destroy"). With speech, it is important to also use aliasing as suggested in [8, 98]. For example, the combination "create' or "destroy" and "new" or "delete" covered nearly all proposals. A few times referents had very close ties for the most agreed-upon gesture. Aliasing would be beneficial here as well. For zoom in the legacy, two-finger zoom won but the pinch and expand were close in proposal frequency, for that case, both gestures should be available.

Nearly every participant in both the speech and the gesture with speech block proposed an utterance that was <action> <direction> or <action> <object> <direction> (2.4). With this observation, we believe that a word spotting algorithm paired with aliasing certain commands together would be sufficient for most speech interaction tasks. This trend was also observed by [9].

When allowed to use both gesture and speech, gestures will typically proceed speech. Some of these gestures will be more generic pointing or turning gestures (screwing in a light bulb) accompanied by an action phrase such as "spin". Deictic gestures are more common when speech is allowed (2.5). The exception is that select had nearly all deictic gestures.

When developing a recognizer system for gesture and speech inputs the timing windows of co-occurring gestures and speech should be considered. When establishing time windows for speech centering the window around ~130 milliseconds after gesture imitation would be beneficial. It should also be noted that each channel provides inputs that are disambiguated with the other channel. Seen in the pointing gesture paired with "delete" or "new" command.

The gestures proposed in this study can be implemented using the sensors built into consumer available AR-HMDs using either the stock hand tracking application program interface or the raw video stream. We found that tracking a few points (e.g., index tip, thumb tip, thumb base) was sufficient for direct manipulations and allowed for variations in the count of fingers used while gesturing. We recommend aliasing gestures across aliasing manipulative gestures across hand positions (open hand, pinch, grab, index only) based on the axis of movement. We also recommend that each one-handed interaction has a symmetric bi-manual version (i.e., one-hand push with two-hand push). While bi-manual gestures were not the most common interaction proposal in this study, other research suggests that with larger objects users opt for larger gestures [73, 74].

2.3.9 Limitations of the Study

While the findings presented here are important, this study has limitations. The environment presented uses one virtual object at a time. While this was by design, it is not clear if the findings will transfer into more complex environments (e.g., Lego-like applications) where object selection is necessary before a command it given. The design choice to ask the referent by using text in the virtual environment, while not uncommon, may have primed some of the participants' speech. Future work will address some of these limitations.

2.3.10 Conclusion

This is one of the first studies to test each of these input modalities independently in a within-subject design allowing us to take a more granular approach to the analysis of co-occurring gesture and speech usage within this environment. Due to that approach we are able to discuss the individual and joint strengths of each modality have been examined and suggestions have been made for both the unimodal and multimodal usage of these modes of interaction. This work extends the work of many linguists [53–55], and the work of computer scientists [59–61, 95] into AR-HMD building environments by examining the syntax patterns of co-occurring speech and gestures as compared to speech alone. We have shown that the timing between co-occurring manipulative gestures and speech in AR-HMD environments follows the same trend as found in studies using other types of

gestures. This finding can be leveraged to create better recognizer systems as well as more natural human-centric interfaces. This study presents a set of user derived ego-centric gestures for use in AR building environments. These ego-centric gestures are critical when using a head-mounted camera such as the ones found on most AR devices. We have also found indications that gesturing is used to reduce cognitive effort when determining the direction of a requested rotation.

Future Work

Multiple unanswered questions require further work. For example, would the findings here translate to more complex environments? What if there are multiple users (either in the same room or not) in a shared virtual environment, would this lead to similar findings as human-to-human communications (e.g. [53]). Another future direction is to perform a follow-up study where the users are asked to generate gestures by seeing the movement of the object (with no text in the virtual environment). These questions are still open for any team to further explore. Head position and gaze were not measured in this study because there was only a single object presented at a time. In future work we plan to assess the role of gaze and head position in multi-object environments. Both gaze and head position serve as passive inputs that can improve accuracy in selection and interaction tasks.

2.4 Understanding Gesture and Speech Multimodal Interactions for Manipulation Tasks in Augmented Reality Using Unconstrained Elicitation[6]

2.4.1 Overview

This research establishes a better understanding of the syntax choices in speech interactions and of how speech, gesture, and multimodal gesture and speech interactions are produced by users

[6]Williams, A. S., Ortega, F. (2020). "Understanding Gesture and Speech Multimodal Interactions for Manipulation Tasks in Augmented Reality Using Unconstrained Elicitation". *Proc. ACM Human-Computer Interaction. V4, ISS, Article 202 (November 2020), 21 pages. DOI: https://doi.org/10.1145/3427330*

in unconstrained object manipulation environments using augmented reality. The work presents a multimodal elicitation study conducted with 24 participants. The canonical referents for translation, rotation, and scale were used along with some abstract referents (create, destroy, and select). In this study time windows for gesture and speech multimodal interactions are developed using the start and stop times of gestures and speech as well as the stoke times for gestures. While gestures commonly precede speech by 81 ms we find that the stroke of the gesture is commonly within 10 ms of the start of speech. Indicating that the information content of a gesture and its co-occurring speech are well aligned to each other. Lastly, the trends across the most common proposals for each modality are examined. Showing that the disagreement between proposals is often caused by a variation of hand posture or syntax. Allowing us to present aliasing recommendations to increase the percentage of users' natural interactions captured by future multimodal interactive systems.

2.4.2 Introduction

Establishing impactful unimodal and multimodal interaction techniques for augmented reality (AR) head-mounted displays (HMDs) starts with understanding unconstrained user behavior. Gesture and speech show promise as the inputs that will be well suited for use in AR-HMDs. Both of these modalities can be tracked with the sensors that come standard on most consumer-available AR-HMDs such as the Microsoft Hololens 2. This minimalism is beneficial. When using AR-HMDs people will likely seek to carry as little extra technology as possible.

Gestures and speech have strengths as both unimodal and multimodal inputs [51]. These strengths have not yet fully been examined. Speech has been found well suited for abstract tasks such as multi-object manipulation [107] or selecting a device out of a set of devices [83]. Gestures have been found well suited for direct manipulation [107]. The combination of these modalities can provide a more rich interaction environment than either alone. By understanding the strengths and synergies of these modalities we can better design systems for the end-user.

We can see some of the impacts of new interaction paradigms in the widespread use of multi-touch devices (e.g.,touch screen cell phones) reaching populations that do not commonly use com-

puters but can benefit from the use of technology [108]. Augmented reality is one of the technologies expected to become pervasive in the future, and with that, interactions in AR-HMD environments will become pervasive. Proof AR-HMDs' increased prevalence can been seen in the the United States government's purchase of $100,000$ Microsoft HoloLens 2 units for Army use [109]. There is little standardization for mid-air gestures AR environments [52], the same can be said for speech inputs. Co-occurring gesture and speech interactions, where both gestures and speech are used to convey a message within close temporal proximity of each other, have been analyzed within the context of human to human interaction [53–55], however, the unconstrained generation of these inputs in human-computer interaction (HCI) has been far less commonly examined [51, 88, 91].

This research presents a study in which participants are tasked with interacting with a virtual object both unimodally and multimodally in an optical see-through AR-HMD environment. These interactions were unconstrained. Gestures, speech, and co-occurring gesture and speech interactions were each tested independently. The main goal of this research was to provide insight on speech interactions, with and without gestures, for object manipulation in AR. To provide robust comparisons, unimodal gesture alone interactions were also examined.

The **contributions** of this research include a detailed analysis of these input modalities' interactions and insights into the changes in those interactions when used multimodally as opposed to unimodally are given. Instead of presenting a single consensus set for each modality, we highlight the common proposals, themes across proposals, and the syntax used for speech interactions. Lastly, timing windows based on the phases of a co-occurring gesture and speech interaction are constructed. Showing that the information content of an interaction is closely aligned with the stroke of a gesture. Based on those findings this paper establishes some guidelines for multimodal gesture and speech input development in this emerging area.

Motivation

Interactions with systems should be intuitive [58]. One way of achieving that is by leveraging interaction modalities that we are familiar with. Interpersonal communication is rich with gesture and speech interaction [53]. Communication is formed in both gesture and speech channels si-

multaneously, with each channel impacting the formation of a message by the other channel [54]. Enabling a system to accept gesture and speech as both unimodal and multimodal input channels, is an important step towards creating intuitive augmented reality interaction design.

When participants were given the option to chose modalities, they chose to combine gesture and speech inputs 60% to 70% of the time [60, 67]. This preference can be used to improve recognition [68]. End-users feel that interactions with a system are more natural when they can chose input modalities based on their preference [69, 70]. By leveraging this preference and multimodal inputs, many benefits can be realized. The use of multiple input channels can lead to mutual disambiguation of information lost in the other channel [64–66], as well as lead to less verbose interactions by allowing for two communication channels to send non-redundant information simultaneously [63]. Gesticulation is closely linked to the structure of co-occurring speech, allowing for better error recovery in recognizers [65].

Optical see-trough AR-HMDs (e.g., Magic Leap One and Microsoft Hololens versions 1 & 2) are starting to implement gesture and speech interactions. That said, these interactions could still use much improvement. Some of the interactions implemented seem built to improve recognition accuracy rather than improving user experience. For example, Magic Leap's C gesture is fairly easy to detect (being a static symbolic gesture) but may not be the most intuitive. Often if gesture sets are not designed with an emphasis on recognition they are designed by experts [71]. User-defined gesture sets have been shown to be up to 24% more memorable [72] and to be preferred to expert-designed gesture sets [8].

This work is not on multimodal fusion (or recognition) [57], rather, it is on multimodal interaction, input generation, and design. Nevertheless, the results of our study can be used by researchers working on multimodal fusion. We use participatory design guidelines to work with potential end-users of AR-HMDs to find what inputs within each modality they would instinctively use [8, 51]. The timing information for phases of a multimodal interaction can help tune recognition windows in multimodal fusion systems. The combination of work on elicitation, such as this study, and multimodal fusion will help HCI build systems with more natural interactions. The technological

gap between the feasibility of traditional inputs and gesture with speech inputs is being minimized, soon the later may become more efficient [70]. This work provides information on the top few interaction proposals for each modality, interaction themes across modalities, co-occurring gesture and speech timing information by phase of interaction, and design guidelines on input design for AR building environments.

2.4.3 Previous Work

Gesture Elicitation

Elicitation is a type of study that aims at mapping inputs to emerging technologies through participatory design. The elicited inputs should be discoverable to novice users of systems [8]. A second product of elicitation studies is a better understanding of user behavior. Elicitation studies have shown that upper-body gestures are preferred in whole-body gesture systems [75], and that gestures produced are impacted by the size of the object [73, 74]. Elicitation has seen use for many input domains such as multi-touch surfaces [80, 81], and mobile devices [82], to internet of things use [83].

Elicitation studies typically use a Wizard of Oz (WoZ) experiment design [8, 76]. WoZ experiment design can be used to remove the gulf of execution between the participant and the system by removing the systems input recognizer [8]. In a WoZ elicitation experiment, a participant is shown a command (referent) to execute such as *move down*. The participant generates an input proposal for that referent which causes an experimenter to emulate the recognition of that input. In this work that is changed slightly to allow for better collection of speech results. For the command *move down* in this experiment, a participant was shown a virtual object moving down after which they would be asked to generate a command to produce that effect. By running the study this way we were able to collect inputs for a system that does not have a perfect recognizer or fusion model.

One outcome of an elicitation study is the production of a mapped set of inputs called a consensus set [79, 80]. More useful than a single set of mapped inputs is the observational data that comes from elicitation studies. This includes insight on the formation of inputs, the times surrounding in-

put generation, and trends in user preferences for inputs and input modalities. An example of these extended benefits is the finding that the size of a gesture proposed is impacted by the size of the object shown [73]. This work extends previous gesture elicitation studies in AR [99] by testing the additional modalities of speech alone and multimodal gesture and speech interactions and allowing unconstrained gesture proposals for each referent. Furthermore, the set of interactions presented here shows the top few proposals allowing better interpretation of trends in gesture formation.

Gesture and Speech studies

A large portion of multimodal gesture and speech input studies have been focused on finding ways to combine them using multimodal fusion models [57, 59, 92, 93]. There has also been work on finding the timing windows for co-occurring gesture and speech interactions [9]. Some of this work looks at the usability of constrained sets of inputs such as limited gesture sets [94] or limited speech dictionaries [9]. These types of works look for a better understanding of a combination of the feasibility of inputs, the adaptability of people to constrained inputs, and the implementation or accuracy of fusion models for gesture and speech recognition. These works typically start with live mapped inputs and test usability or accuracy. **The work presented here is very different in that there are no constraints imposed on input proposals, and deliberate efforts were made to remove text based priming in the speech condition**. Participants are invited to generate any input proposal they see fit for the given referent and input modality.

While a few studies look at gesture and speech inputs have examined mid-air gestures [9,43,51, 57, 87, 90], some only looked at a subset of gesturing such as pointing gestures [89, 95], paddling gestures [91], or two dimensional (2D) gestures [88, 89]. The work presented here examines any mid-air gesture and / or utterance that a participant feels is appropriate for a given referent.

This study extends previous works done on multimodal gesture and speech elicitation [43, 51]. This extension is seen in the results reported and the methodology used. A previous study on interactions for computer-aided design program usage on 2d screens tested both gesture and gesture or speech interactions [43]. In that experiment, gestures were tested independently then gesture with optional speech was tested. This is different from our choice to examine each input individually.

In both studies the referents were shown as animations, however, in this study participants were told that they were interacting with a system whereas Khan et al. asked participants to describe the referents to another person via a video chat [43]. The use case of computer-aided design as well as the choice of observing interactions compared to referent descriptions is markedly different, with examples of the referents used there being *extrude surface* or *pan*.

This work also extends the results of a study done on eliciting commands for television-based web browsing [51]. That study used paired elicitation where participants would sit in groups on a couch and propose either gesture, speech, or gesture and speech commands, as compared to the individual elicitation technique used here. That study also only examined the input modalities in a single pass where participants were allowed to produce any command in any modality or a combination of modalities. An important distinction is that referents were shown as text and read aloud by the experimenter in Morris, 2012 [51]. In this study we examine interaction proposals without text prompting.

This work differs from previous gesture and speech elicitation studies in several important ways. This work does not present users with any text when showing referents. Participants are not paired and are asked to produce an input for each modality. This is in comparison to prior works which commonly allows users to chose which modality they use when generating input proposals [48]. This work aims on finding intuitive inputs across the gesture, speech, and co-occurring gesture and speech interactions. This work does not attempt to improve gesture or speech recognition, nor does it attempt to build better multimodal fusion models. It is our hope that these results can be used towards those goals in future studies.

2.4.4 Methods

Pilot Studies

Two versions of this study were run to assess the impact of referent display on proposal generation. The results of these pilot studies were used to inform the methodology decisions made in this experiment. These each used 6 people. In one of the pilot studies, we display the referents as

text on the screen, which is different from our final design. The first pilot study's design is comparable to [8, 51]. In the second pilot study, we displayed the referent by showing the participants an animation of the intended effect of the interaction they would propose. The second pilot study's design is comparable to [43]. Both the pilot studies and this study tested the same input modalities, those being, gesture and speech, speech alone, and gesture alone.

In the first pilot study, there was evidence that text referents primed speech production. If the referent was *move right* the utterance was commonly "move right". This effect was more pronounced for translations, rotations had more variance in proposals but still showed signs of biasing. Repeating referents when producing speech proposals, such as saying "new tab" for the referent *new tab*, can be seen in the results of Morris, 2012 [51]. When the referents were shown as animations in the second pilot study, people would often mirror that animation in the gesture they produced. These mirrored gestures were often direct manipulations which are not uncommon in gesture interfaces [46], however, when designing inputs that priming could be problematic. The effect animations biasing gestures can be seen in the study done by Khan et al. 2019 [43], such as a *pan* gesture that mirrors the motion of the animation used.

This study's goal was to understand user speech behavior both alone and when co-occurring with gestures. With that in mind, we have chosen to show the referents as an animation. The only text shown to the participant was the input modality requested (e.g. "gesture only", "speech only", "gesture and speech"). This will allow us to have more robust speech results than when showing a text based referent. Another choice in elicitation methodology used in this experiment was to not have think aloud protocol as seen in [8]. The process of thinking out loud while generating speech proposals would confound the results, making speech data less reliable.

Methodology

This study was run as a within-subjects (i.e. repeated measures) elicitation study. The goal of this work was to gain a better understanding of the production of gestures, speech, and co-occurring gestures and speech when interacting with three-dimensional (3D) objects in an optical see-through AR-HMD. Participants were asked to generate proposals for gesture alone, speech

alone, and multimodal gesture and speech interactions. These input modalities were presented in a counterbalanced order. Within each input, participants were asked to generate an interaction proposal for each referent. Meaning that a participant may be assigned the speech input modality first, then be asked to generate a speech proposal for each referent before progressing to either the gesture or gesture and speech condition. Referents were displayed in random order with each occurring once per input modality. The experimental setup is illustrated in Figure 2.6. Participants were told that they were guessing the interaction that someone in a different room was using to execute the referent they were presented with. A single referent sequence was a blank screen, a cube appearing, a 2-second pause, the cube playing an animation of the referent, then the participant proposing their input. The animation playing first removes the notion that the participant is directly interacting with the system. However, their belief that someone else is interacting with this system in a separate room, and the onscreen gesture aids (described later), caused the user to feel that this was a live system.

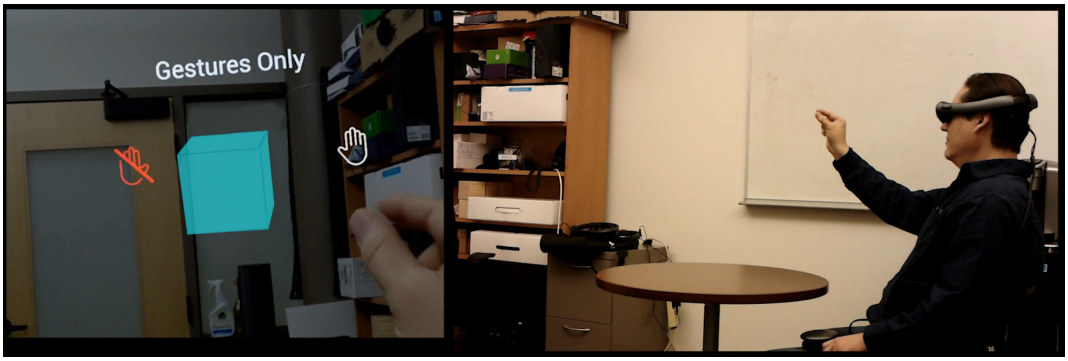


Figure 2.6: Experimental Set up: Left, participant view, Right: participant

The referents (i.e. actions) that were used included the canonical manipulations (i.e. selection, rotation, positioning) found in [46] and the interactions that would be commonly used in a 3D manipulation or building task. They include translation and rotation on each axis, scaling, selection, and the creation or deletion of an object. This study looks at the use case of a 3D environment such as an architecture application, where objects must be manipulated and placed within that environment. This can be extended into interactive learning environments or data visualization en-

vironments where manipulating virtual content can provide better learning outcomes [110]. Most optical see-through AR-HMDs (e.g., Magic Leap One) and some VR-HMDs (e.g., Oculus Quest) have built in ego-centric sensors. With that in mind, the gestures in this study were analyzed by viewing the ego-centric interactions within the environment.

The metrics used for gesture proposal interpretation are Agreement Rate ($\mathcal{AR}$)[7], co-agreement rate ($\mathcal{CAR}$) , and the (V_{rd}) significance test [8, 49, 77]. $\mathcal{AR}$ is the proportion of proposals in agreement over the total possible proposals pairs in agreement. High $\mathcal{AR}$ can be interpreted as more consensus among participants in the proposals generated for a given referent. This metric is used at the referent level meaning that a given proposal will not have an associated $\mathcal{AR}$ but a referent will. Based on distributions of $\mathcal{AR}$ over various sample sizes participants an $\mathcal{AR}$ of 0.3 has been said to indicate high agreement given our N of 24 [49]. The V_{rd} is a test of the difference in agreement rates between k referents. A low p-value indicates that there is a difference between the tested referents. The $\mathcal{CAR}$ can be seen as the percent of participants that agree on proposals for k referents. Fleiss' Kappa and the associated chance agreement term are used to justify using an $\mathcal{AR}$ of 0.3 as high [50].

For speech proposal analysis the consensus-distinct ratio ($\mathcal{CDR}$) and max-consensus ($\mathcal{MC}$) were used [51]. The $\mathcal{CDR}$ is the percent of matching proposals that have been suggested by more than a recommended baseline of two participants out of all the proposals for a given referent [51]. $\mathcal{MC}$ is equal to the percent of participants proposing the top-ranking proposal. The combination of these metrics can be used to see the peak and spread of speech proposals.

Participants

The study consisted of 24 volunteers (10 Female, 14 Male). Participants were recruited using emails and word of mouth. Participants were 18 - 46 years old (Mean = 25, SD = 6.9). Six participants had less than half an hour of previous AR-HMD usage experience, the other participants had no prior device usage. All participants reported normal or corrected to normal vision. Five

[7]Please note that agreement rate $\mathcal{AR}$ uses a different font to avoid confusion with AR for augmented reality.

participants reported being left-handed. Five participants reported weekly use of VR. Only one of 2 of those participants used VR more than 5 hours weekly (5 hours, 10 hours), the rest were 1-3 hours weekly.

Procedure

For each session participants started by completing the informed consent and demographic questionnaire. That questionnaire asked about prior device usage (AR, VR, multi-touch), age, handedness, vision, and gender. A two-minute instruction video was shown describing the experiment after which the participant could ask the experimenter questions. During the video, they were told that any utterance or gesture either one-handed or two-handed, produced was acceptable. The participant would then don the AR-HMD and complete a practice trial for each input modality. During the practice trials, the participant could ask any questions they had and adjust the device. Participants were also alerted to the devices gesture recognition aid shown (Figure 2.6) during the practice. This aid was an image of the outline of a left and right hand. The hands were white when a participant's corresponding hand was inside the device's gesture sensing range. They would be red with a line through them when the participant's corresponding hand was outside of the device's recognition range. This aid was provided to help prompt participants to generate gesture proposals that could be used in AR-HMDs as well as to add more immersion to the interactions with the object in the experiment. As this was a WoZ study, the aid was only adding realism to the task, no gestures were recognized.

The referents were shown as animations (showing the object then moving it left over 2 seconds for the referent *move left*). No text was shown to the user. For three referents animations that were not basic movements had to be shown. For the *create* and *delete* referents particle effects of an object appearing or disappearing over two seconds were used. For the *select* referent, the object was highlighted by increasing its hue and adding a light outline. Each referent was presented as a cube rendered 50cm in front of a user's display. The modality to use for the proposals was shown as text above the cube. The experimenter would trigger the loading of the next referent a few seconds after a proposal was generated by the participant. The new referent would always appear

in the center of the participant's display, stay there for 2 seconds, then execute the animation for the referent.

Apparatus

This experiment was conducted using a Magic Leap One optical see-through AR-HMD. The WoZ system was developed in Unreal Engine 4.23.0. A Windows 10 professional computer with an Intel i9-9900k 3.6GHz processor and an Nvidia RTX 2080Ti graphics card was used for development. Data were recorded on the Magic Leap One. A GoPro hero 7 black was used to record an ego-centric view of the interactions for analysis. A 4k camera was used to record an exo-centric view of the interactions as a backup to the GoPro.

2.4.5 Results

Gestures Proposals

Gestures from the unimodal gesture block The average $\mathcal{AR}$ observed for the gesture block was 0.302 with $\kappa_F = .257$. Given our sample size of 24 and the low chance agreement term ($p_e = .052$) used in Fleiss' Kappa coefficient we consider rates above 0.3 as high levels of consensus [49, 50]. Agreement rates are shown in Table 2.9.

Table 2.9: Agreement rates per referent by block

	Create	Delete	Enlarge	Move Away	Move Down	Move Left	Move Right	Move Towards	Move Up	Pitch Down	Pitch Up	Roll C	Roll CC	Select	Shrink	Yaw Left	Yaw Right
Gesture	0.21	0.11	0.28	0.37	0.38	0.49	0.44	0.28	0.49	0.16	0.28	0.56	0.39	0.09	0.14	0.25	0.22
Gesture and Speech	0.09	0.05	0.18	0.50	0.29	0.33	0.32	0.34	0.35	0.19	0.22	0.28	0.45	0.09	0.14	0.15	0.25

Legend: C: clockwise, CC: counterclockwise, Highlighted cells have high agreement

The effect of referent type on agreement rates was observed to be significant ($V_{rd(16,N=408)} = 510.342, p = .001$). High agreement was found for each of the translation referents except *move*

away, and for both the *roll clockwise* and *roll counterclockwise* referents (Table 2.9). The highest $\mathcal{AR}$ was found in the *roll clockwise* referent ($\mathcal{AR}_{roll\ clockwise} = .56$). A mapping of the frequency of gesture proposals with more than three participants suggesting them and the corresponding referents can be seen in Figure 2.7. The gestures from the gesture block have "G" next to the referent name.

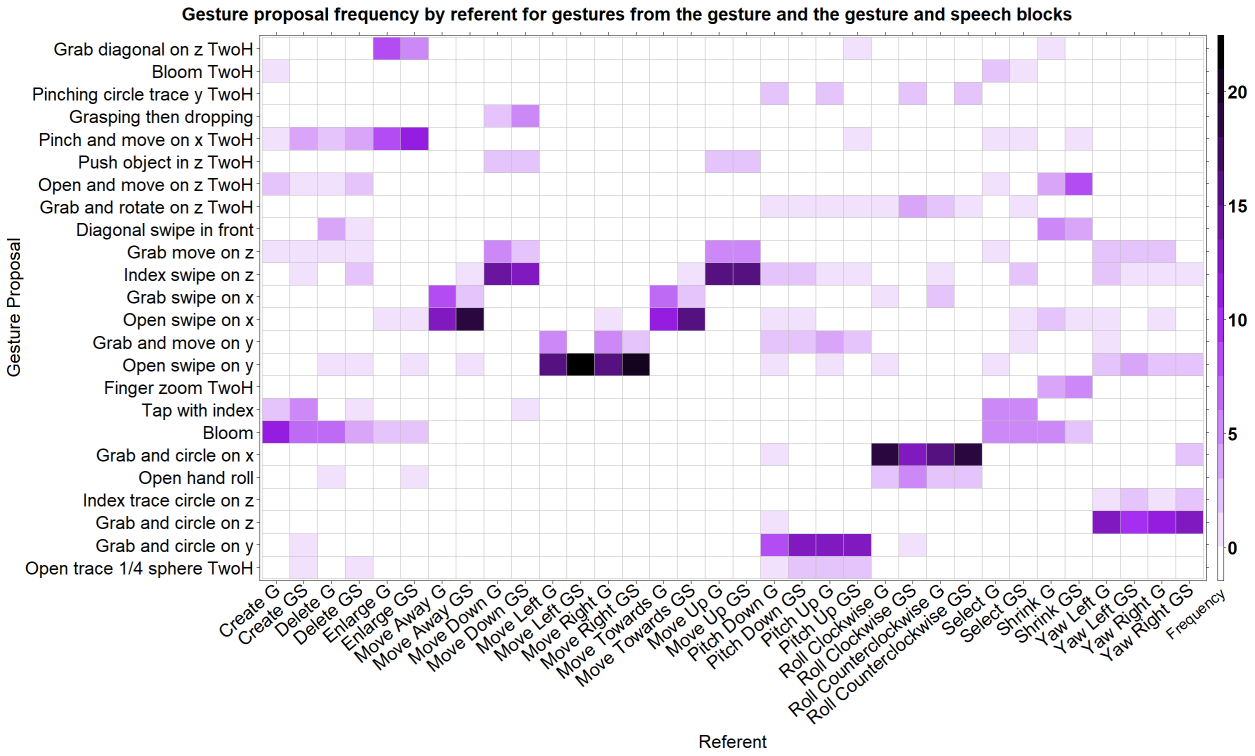


Figure 2.7: Gesture proposal frequency by referent for gestures from the gesture and the gesture and speech blocks

Legend: G: Gesture Block, GS: Gesture and Speech block, TwoH: Two handed gesture, Open: fingers open, Grab: hand closed, Pinch: two or three finger pinching, z: up, x: forward, y: side

The more abstract referents, *Create*, *Delete*, and even *select* exhibited low agreement rates ($\mathcal{AR}_{shrink} = .14$, $\mathcal{AR}_{delete} = .11$, $\mathcal{AR}_{Select} = .09$). This is mostly due to disagreement between proposals shown by an increase in the count of colored cells in Figure 2.7. Common hand poses and movements are shown in Figure 2.8. *Select* had low $\mathcal{AR}$ due to participants having a difficult time interpreting the referent animation. *select*'s animation showed the cube normally (left side of Figure 2.6) then gradually becoming highlighted by reducing the hue after a 2-second delay. In

pilot tests on the *select* referent we attempted other visualizations such as bouncing, or an arrow appearing and pointing at the cube. These animations primed the speech and gesture produced. The highlight animation had the highest rate of participants guess what it was, but that rate was still fairly low.

The translation referents (*up, down, left, right, away*, and *move away*) had high gesture agreement among participants ($\mathcal{AR}_{translations} = .432$). Among these translational referents, the direction of motion displayed a significant effect on agreement rates ($V_{rd(5,N=144)} = 52.765, p < .001$). A significant difference in agreement was observed for referents *towards* and *away* ($V_{rd(1,N=48)} = 9.921, p = .001$). *Roll clockwise* and *roll counterclockwise* had high $\mathcal{AR}$ with an average ($\mathcal{AR}_{roll} = .475$). This was higher than the average $\mathcal{AR}$ for all the rotational referents ($\mathcal{AR}_{rotations} = .31$) which drops to ($\mathcal{AR}_{rotations\ without\ roll} = .23$) when roll is removed. We believe that participants may not have had much experience with altering the pitch or yaw of virtual objects and this is reflected with the low $\mathcal{AR}$. The excepting being roll manipulations, which seem more common with objects like clock hands moving that way, inflating their $\mathcal{AR}$.

There was low $\mathcal{AR}$ for *shrink* and *expand*, which is surprising due to the prevalence of touch-screen phones and near-daily use of the two-finger zoom-in and zoom-out commands. Those gestures occurred with some frequency, however, there were a high number of two-handed comparable gestures proposed (Figure 2.7). For these people would pinch either corner and pull or push their hands away or towards other either diagonally or horizontally.

The heatmap in Figure 2.7 helps show the trends among gesture proposals, darker colors indicate more proposals. The gestures mapped are all reversible gestures meaning a movement in the opposite direction is the mirror of the gesture. An example of this is seen in the gesture for *move up* which was a palm up push up where *move down* was a palm down push down. The referents *move left* and *move right* had very few different proposals indicating high agreement on the appropriate gesture. Whereas, referents like *select* had a high range of proposals given. When examining the plot horizontally by proposal instead of vertically by referent trends in how participants map the same gesture to multiple actions are seen. For example, an open hand swipe either left or right

was used for 9 referents. The uses make sense, a quick swipe from right to left could be seen as deleting an object, or touching the side of an object and moving left or right would change its yaw. The "Bloom" gesture was used for every abstract referent. The variations present in some manipulations were only in the pose of the hand, or the number of hands, but not the motion of the gesture. *Move up* had three common proposals with each centering around some sort of grab and a movement on the z-axis.

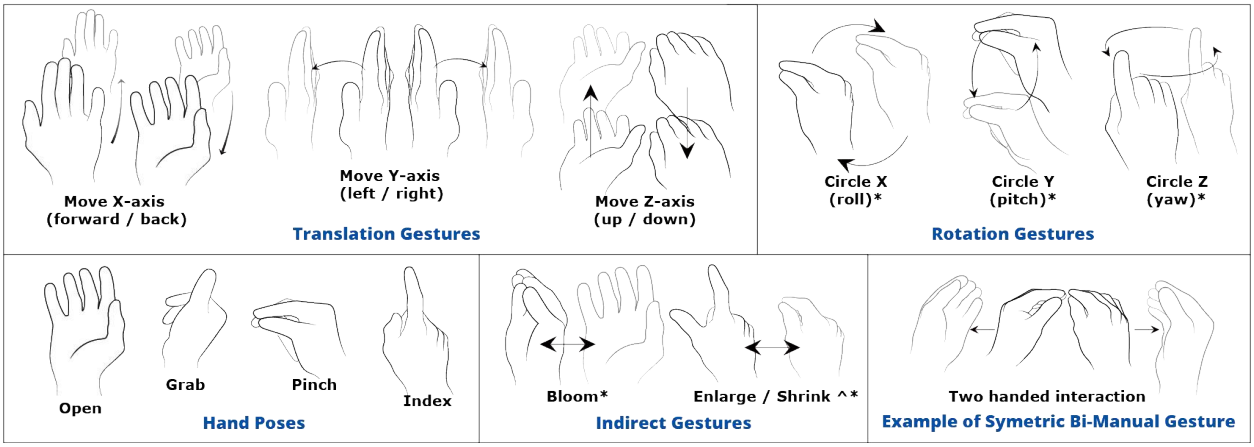


Figure 2.8: Hand pose examples, two handed gesture example, and common gestures by category of movement or type of gesture

Legend: *: reversible gesture, $\wedge$: commonly two handed, z: up/down, x: forward/back, y: left/right

Gestures from the multimodal gestures and speech block The results for the gesture proposals from the gesture and speech are very similar to the gestures from the gesture alone block. By comparing columns with the matching referent names (e.g. *create G* and *create GS*), an image of the differences of proposals across these blocks can be drawn. The overall agreement rate observed for the gestures in the gesture and speech block was 0.247 with $\kappa_F = .218$. The low chance agreement term ($p_e = .037$) used in Fleiss' Kappa coefficient indicates an agreement beyond chance [50], allowing us to consider $\mathcal{AR}$ rates above 0.3 as high [49]. The agreement rates for each referent are shown in Table 2.9.

The effect of referent type on agreement rates was observed to be significant ($V_{rd(16,N=408)} = 904.091, p = .001$). High agreement was found for each of the translation referents except *Move Down* $\mathcal{AR}_{Move\,Down} = .29$. This was caused by an increase in the number of "drop" gesture proposals. *Roll counterclockwise* also exhibited high $\mathcal{AR}$ ($\mathcal{AR}_{Roll\,counterclockwise} = .45$) (Figure 2.9). The highest $\mathcal{AR}$ was found in the *Move Away* referent ($\mathcal{AR}_{Move\,Away} = .5$). A mapping of the frequency of the top gesture proposals and the corresponding referents can be seen in Figure 2.7.

The abstract referents *Create*, *Delete*, and *select* exhibited low agreement rates ($\mathcal{AR}_{Create} = .09$, $\mathcal{AR}_{Delete} = .05$, $\mathcal{AR}_{Select} = .09$). This is mostly due to disagreement between proposals shown by an increase in the count of colored cells in Figure 2.7. As in the gesture block, *select* had low $\mathcal{AR}$ due to participants having difficulties interpreting the referent's animation. The translation referents (*up, down, left, right, away*, and *move away*) had high gesture agreement (average $\mathcal{AR} = .355$). A significant disparity was observed for referents *roll clockwise* and *roll counterclockwise* ($V_{rd(1,N=48)} = 59.522, p = .001$). *Roll clockwise* and *roll counterclockwise* had high $\mathcal{AR}$ with an average of ($\mathcal{AR}_{Roll} = .475$. This was higher than the average $\mathcal{AR}$ for all the rotational referents ($\mathcal{AR}_{Rotations} = .31$) which drops to ($\mathcal{AR}_{Rotations\,without\,roll} = .23$) when roll is removed. We believe that participants may not have had much experience with altering the pitch or yaw of virtual objects and this is reflected with the low $\mathcal{AR}$. As in the gesture block the scale referents had low $\mathcal{AR}_{Shrink,Enlarge} = .18, .14$.

The bulk of the gestures shown in Figure 2.7 are direct manipulation gestures. Translations are concentrated in a few gestures where rotations are spread across more proposals. Even so, most rotation proposals involved tracing or moving a participant's hand in a circle. In the case of most of the referents, there was an increased spread of gesture proposals in the gesture and speech block. This was not the case for every referent, some such as *move left* and *roll counterclockwise* have a decreased number of proposals in the gesture and speech block. Largely the gestures used did not change drastically between the two blocks.

Speech Proposals

Displaying the referent in elicitation studies [37] and reading the referent aloud in gesture and speech elicitation studies [51] both have precedence. These practices can prime the utterances proposed. When interpreting these results remember that neither think out-loud protocol nor text was used for referents. The participant only saw an animation of the referent being executed. When analyzing speech proposals we have dropped the object specifier to remove a level of increased proposal complexity. We believe that if an object is already selected, using the command "Move the cube right" and "move right" could be reasonably considered the same, the exception being the *select* referent.

Table 2.10: Frequency of syntax format by block

	<action>	<action> <direction>	<action> <object> <direction>	<action> <object>	<direction>
Speech	28.19%	47.06%	14.22%	9.31%	1.23%
Gesture and speech	38.48%	39.95%	12.99%	6.86%	1.72%

Speech from the unimodal speech block While were told that any utterance or sentence was acceptable, they primarily stuck to <action> <direction> or <action> <direction> syntax structure. The rates for syntax are found in table 2.10. The difference between <action> <direction> and <action> <object> <direction> was only a descriptive specifier of the object (e.g. “cube”). The <action> and <direction> words were the same as found when no specifier was used (e.g. “move the cube left” would be “move left”).

The $\mathcal{MC}$ and $\mathcal{CDR}$ for this block are shown in Figure 2.11. Note that $\mathcal{MC}$ is equal to the percentage of participants proposing the top proposal per referent, shown in the "Top proposal" column in Table 2.11. *Yaw* referents had some of the highest $\mathcal{CDR}$ indicating high disagreement among participants on the utterances proposed ($\mathcal{CDR}_{Yaw\ left, Yaw\ right} = .62, .78$). *Delete* also had a high amount of disagreement among proposals ($\mathcal{CDR}_{Delete} = .57$). Both *create* and *shrink* had

low $\mathcal{CDR}$ ($\mathcal{CDR}_{Create,\,Shrink} = .18, .25$). Low $\mathcal{CDR}$ means that most participants grouped around the top proposals. The rest of the referents hold moderate disagreement values.

The highest $\mathcal{MC}$ value belongs to *move up* ($\mathcal{MC}_{move\,up} = .54$). Most participants proposed either "Move up" (54.17%) or "go up" (12.5%). The full list of each referent's top two proposals and the percent of participants proposing them can be seen in Table 2.11. For the translational referents "move" was used as the <action> command in either the top or second place proposal. *Move down* ($\mathcal{MC}_{move\,down} = 33.33\%$), which had "drop" as the top proposal, was the only translational referent that did not have "move" in it. The second-place proposal for *move down* was "move down" (29.17% proposed). The referents for *move up*, *left*, and *right* all had the directional term (up, left, right, down) included. *Move towards* and *move away* had either towards, and forward, or away, and back proposed as the <direction> term. This indicates that aliasing "away" with "back", and "towards" with "forward". Aliasing commands has been suggested as being beneficial when dealing with unimodal speech [8, 51]. Note that these terms are reversible, which was a common trend with most opposite proposals (e.g. "appear", "disappear").

For the rotational referents (*pitch*, *roll*, *yaw*) the average $\mathcal{MC}$ was 24.31% which is lower than the translations average $\mathcal{MC}$ of 35.42. For each rotation the action was specified by either "spin" or "rotate" in all of the top proposals by participants (Table 2.11). This is not unexpected, the terms "roll", "pitch", and "yaw" are uncommon in most fields. *Pitch* has the most unique mapping of proposals commonly "towards", "away" for *pitch up* and "back" for *pitch down*. *Roll* and *yaw* have the terms "left" and "right" for directions. We believe that this ambiguity is solved by adding gestures to indicate the "spin" direction, or by an expert assigning speech commands such as "spin clockwise" in the *roll clockwise*.

The referents *create* and *delete* had single word commands for the top and second place proposals as well as some of the higher $\mathcal{MC}$ found ($\mathcal{MC}_{create,delete} = 41.67\%, 50\%$). The top proposals were "appear" and "disppear". These proposals could be considered similar to the reversible gestures found in this study and others [8,99]. "Create" appeared as a second place proposal (20.83%) and "delete" was a third place proposal (12.5%). *Shrink* ($\mathcal{MC}_{shrink} = 45.83\%$) also had a high

Table 2.11: Speech proposals for the speech from the speech block and the speech from the gesture and speech block

	Speech from the speech block					Speech from the gesture and speech block				
Referent	Top proposal	$\mathcal{MC}$	2nd place	$\mathcal{MC}$	$\mathcal{CDR}$	Top proposal	$\mathcal{MC}$	2nd place	$\mathcal{MC}$	$\mathcal{CDR}$
Create	appear	41.67%	create	20.83%	0.18	appear	33.33%	create	29.17%	0.18
Delete	disappear	50%	remove	16.67%	0.57	disappear	54.17%	make disappear	12.5%	0.33
Enlarge	enlarge	37.5%	grow	16.67%	0.36	enlarge	25%	grow	20.83%	0.56
Move Away	move back	25%	move away	12.5%	0.38	move back	16.67%	push away	16.67%	0.64
Move Down	drop	33.33%	move down	29.17%	0.44	drop	29.17%	move down	16.67%	0.46
Move Left	move left	37.5%	slide left	20.83%	0.44	move left	25%	slide left	16.67%	0.2
Move Right	move right	41.67%	slide right	20.83%	0.44	move right	20.83%	slide right	20.83%	0.33
Move Towards	move forward	20.83%	move towards	12.5%	0.36	move forward	16.67%	move towards	12.5%	0.43
Move Up	move up	54.17%	go up	12.5%	0.33	move up	41.67%	go up	8.33%	0.33
Pitch Down	rotate	20.83%	rotate towards	16.67%	0.46	spin forward	20.83%	rotate towards	16.67%	0.6
Pitch Up	rotate away	16.67%	spin backward	12.5%	0.5	spin back	16.67%	rotate	12.5%	0.43
Roll C	spin right	20.83%	rotate	16.67%	0.5	rotate	20.83%	rotate right	16.67%	0.36
Roll CC	spin left	25%	rotate left	20.83%	0.4	spin left	25%	rotate	16.67%	0.23
Select	glow	20.83%	highlight	20.83%	0.55	change	25%	glow	25%	0.36
Shrink	shrink	45.83%	minimize	8.33%	0.25	shrink	41.67%	make smaller	8.33%	0.23
Yaw Left	spin left	33.33%	rotate	16.67%	0.62	spin	29.17%	rotate left	16.67%	0.36
Yaw Right	spin right	29.17%	rotate	12.5%	0.78	rotate right	20.83%	spin	20.83%	0.6

Legend: C: Clockwise, CC: Counterclockwise, $\mathcal{MC}$: Max-Consensus, $\mathcal{CDR}$: Consensus-Distinct Ratio

agreement between participants. As did *enlarge* ($\mathcal{MC}_{enlarge} = 37.5\%$). *Select*, with its difficulties in animating had low agreement and high spread of proposals ($\mathcal{CDR}, \mathcal{MC}_{select} = .55, 21\%$).

Speech from the multimodal gesture and speech block A chi-square test of independence showed that there was a significant association between the block and syntax choice ($X^2(4, N = 408) = 10.928, p = 0.027$). Participants used a higher rate of <action> only syntax than found in unimodal speech. <Action> <direction> syntax use was reduced by 7.11%. The rates for the syntax are found in Table 2.10. Both of the syntax structures that used an <object> specifier were lower in this block. Most often when an object would have been specified it was replaced by a gesture indicating the object. This gesture was often reaching out and grabbing or another type of direct manipulation.

The average $\mathcal{MC}$ for the translational referents decreased by 10.33% from the speech block (Figure 2.11). This was due to more participants using the <action> syntax. The $\mathcal{CDR}$ did increase in the translational referents as well. Participants had less agreement on the appropriate proposal and the spread of proposals was wider. Even with the differences in syntax use between blocks, the top choice proposals remained the same.

The rotational average $\mathcal{MC}$ only decreased by 2%, the $\mathcal{CDR}$ decreased by 0.113. This means that while agreement on the top choice proposal was negligibly impacted between blocks, the spread of proposals given in the gesture and speech block for rotations was narrower than in the speech block. Most of the top choice proposals for translations changed between the two blocks (Table 2.11). Some switched from using "spin" to "rotate" or vice versa. As an example, the proposal for *yaw right* switched from "spin" to "rotate" while the top proposal for *roll clockwise* did the opposite. We take this to mean that the words "rotate" and "spin" are without a clear mapping in participants' minds. For translations gesturing removes much of the ambiguity by allowing for a physical motion to indicate the intended rotation direction.

Most proposals remained the same between the two blocks with slightly different $\mathcal{MC}$ rates. There was a shift in *create* from the top choice proposal of "appear" from ($\mathcal{MC}_{Create} = 41.67$) to ($\mathcal{MC}_{create} = 33.33$) in the gesture and speech block. This is captured in an increase of 8.34%

in the second choice proposal in the gesture and speech block. *Delete* was mostly unchanged in top proposals but did have a decreased $\mathcal{CDR}$ ($\mathcal{CDR}_{Delete} = .33$). Meaning there were less distinct proposals made. *Enlarge* had a lower $\mathcal{MC}$ and higher $\mathcal{CDR}$ in the gesture and speech block ($\mathcal{MC}, \mathcal{CDR}_{enlarge} = 37.5\%, .56$).

Co-occurring gestures and speech proposals

When looking at pairings of speech and gesture proposals in the gesture and speech block the agreement rates fall drastically due to the highly nuanced nature of speech. Individually each modality had referents that experienced high levels of agreement. For gestures refer to Figure 2.7 and Table 2.9. For speech consensus refer to Table 2.11. We feel that matching common syntax structure with gestures when looking at multimodal gesture and speech interactions is more beneficial than observing the pairing of utterances with gesture proposals. The speech syntax by block is shown in Table 2.10. Gesturing remains consistent in both conditions indicated by a high p-value in a chi-square test ($X^2(49, N = 408) = 10.928, p < 0.247$) (Comparing G and GS in Figure 2.7). The same is true of speech (Compare S and GS in Table2.11). This is beneficial in a few ways. In the case of translations and scaling it allows each input to serve as a back up to the other. Allowing for mutual disambiguation as found by [64]. In the case of rotations, the gesture provides context on the direction of rotation while the speech was commonly "spin" and a direction. With abstract commands, the same gesture, a "bloom" gesture, was found for multiple referents. In those cases, speech allows interpretation of which command is being executed with the gesture.

Timing of co-occurring gestures and speech In the gesture and speech block the time windows of phases of a co-occurring gesture and speech interaction were measured based on the time of gesture initiation. These were collected from videos of the experiment and hand-annotated. The phases used to describe interactions are gesture initiation, stroke start, speech start, stroke stop, and speech stop. These are taken from McNeil's segmentation of co-occurring gesture and speech interactions [53]. The gesture start is the first perceptible movement made by someone. Speech

start is the first perceptible sound being made. For both of those if a false start was found it was discarded and the time of the next movement was taken. As an example, if a participant said "Ummm" then later said "move", the time of "move" was used. A stroke is considered to be the segment of a gesture that holds the information content of the gesture, as well as the peak of effort in that gesture [53]. Gesture stroke was found by measuring the time of the first visible change in the direction of the gesture. The stroke stop was the last change in direction and was found by reversing from the end of a gesture. A full gesture interaction would look like someone starting to move their hand in preparation for a stroke (gesture start), starting a meaningful gesture (stroke start), then ending the gesture (stroke stop). The hand moves up in preparation, pushing the object forward, then retracts to its initial state.

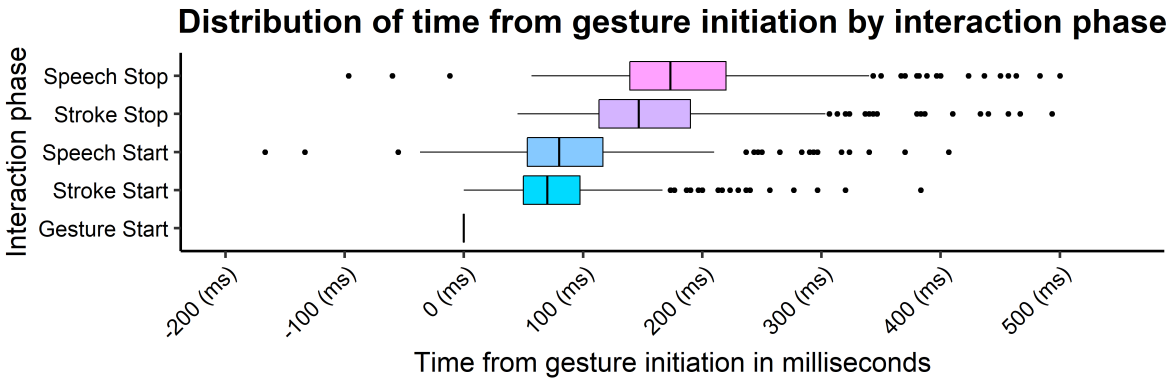


Figure 2.9: Distribution of time from gesture initiation by interaction phase

Shapiro-Wilks tests show that the time information took a non-normal distribution of each of the phases at $(p < .001)$. Bonferroni adjusted Wilcoxon rank-sum pairwise comparisons indicate that the distributions for each time are shifted (i.e., different). The p-values were $(p < 0.001)$ in each comparison except between "stroke start" and "speech start" which was $(p = 0.03)$. The descriptive statistics for times by the phase of interaction are shown in Table 2.12.

We find that in this experiment speech nearly always occurs after a gesture is started (Figure 2.9). The difference in start time is around 81.667 ms. Importantly, the information content of the gesture, the stroke, starts commonly 90.872 ms after gesture start (Table 2.12). This means that

by watching a gesture's changes in direction, we can predict when speech will occur, and when a meaningful message is communicated. Strokes were found to end before speech 23.739 ms. The total interaction from start to finish was typically 187.566 ms. Most speech proposals were only 2 words so this relatively short interaction time makes sense.

Table 2.12: Time from gesture start for phases of an interaction in milliseconds

	Gesture Start	Stroke Start	Speech Start	Stroke Stop	Speech Stop
Mean	0	90.872	81.667	163.827	187.566
Standard Deviation	0	70.548	54.743	78.742	80.064
Standard Error	0	3.493	2.710	3.898	3.964

These results are similar to previous work [9, 95], though slightly quicker and more granular. These results expand time windows from being formed for pointing gestures only [95], and show that these time windows follow similar patterns for deictic and manipulative gestures. They also show that gesture and speech interactions in AR-HMDS have similar timings [100] and patterns of occurrence [101] as in other environments.

2.4.6 Discussion

The hand positions found here were similar to the ones observed by Piumsomboon et al. [99]. The gesture proposals were commonly single-handed. This is similar to findings on multi-touch surfaces [103–105] and mid-air full-body studies [37]. For manipulations users often interacted based on that actions real-world corollary. This is evident in the translation gestures which were predominantly some form of directly pushing the surface of the object. This theme of interaction was seen with manipulation gestures in previous work [99]. We speculate that the similarities in proposals were due to the object being rendered in the participant's real-world view by use of optical see-through AR. With that, users would interact based on their interpretations of naïve physics when possible [111]. This was mostly true for rotations which were accomplished by either grabbing some part of the object and moving their hand in circle motions as also seen in

Piumsomboon et al.'s study [99]. The exception to these similarities is in the occurrence of "index extended" circular motions as an indirect gesture.

Scaling was often a two-handed pinch and drag gesture which was more common than touch screen "zoom in" and "zoom out" gestures. Grabbing the corners or sides of an object would correspond with how a mental model of a stretchable object would be manipulated. This gesture was seen for scaling on an axis in [99]. Similarities in gesture proposals between these studies start disappearing as the referents become more abstract. This can be seen when comparing the proposals for *delete* which was a "grasping" gesture in other work [99] and a "bloom" gesture here.

That most of these gesture proposals extend across two studies and two-time points is a strong indication that these gestures and hand poses should be candidates for inclusion in future AR interaction systems. This study did not ask participants to reserve proposals for a single interaction (i.e., a bloom could be used for *create* and for *select*). Redundantly mapped proposals showed up more in the abstract referents. In the work of Piumsomboon et al. participants were asked to refrain from redundantly mapping inputs [99]. The similarities of proposals between these works show that requiring unique interactions may not have greatly impacted many of the gesture proposals [99]. An interesting, redundantly mapped gesture was the "index swipe" which was used for both *yaw* and *move up/down*.

We feel that the combination of high levels of agreement for translations in the gesture block and the tendency to have more unique proposals given in the gesture and speech block indicate that unimodal gestures are well suited for object manipulations. While rotations had a high number of single-hand "grab and rotate" gestures, many were indirect manipulations using a index finger and tracing a circle. For these, a non-isomorphic gesture seem well suited. The most agreed-upon proposals for manipulations were all reversible gestures. Indicating a preference for reversible gestures which mirrors previous work [8, 99].

Some of these direct manipulations were implemented and tested against a gesture+speech interface in the work of Piumsomboon et al. [107]. The findings were similar to the user stated expectations observed here. When specific degrees or units were needed participants indicated a

preference for speech. For most basic or single object manipulations, gesture seemed preferred across both studies [107]. Peoples' preference for multimodal interactions typically increases as a task's cognitive load increases [112] or the task's complexity increases [107]. We expect that if more complex referents were used the user stated preference for multimodal interactions would have been higher.

Gestures showed less usability for the *create* and *delete* referents. Speech had more clarity in these cases with common utterance being "appear" and "disappear". Gesture proposals for abstract referents were consistently the "bloom" gesture, which was proposed for many referents, and thus hard to interpret without additional context. Speech show more promise for use with abstract commands and conceptually difficult actions that do not map well to a user's mental model. An example would be opening a new browser window, which was not tested here. Speech proposals for both *create* and *delete* had high agreement, emphasizing this strength.

When used together gestures and speech provide different benefits based on the type of referent being executed. For translations and scaling this was commonly redundancy, which allows for error correction in a recognizer system. For rotations, this pairing allows a clear communication of the desire to rotate then clarifying the direction with a co-occurring gesture. This allows for intuitive interactions with mutual disambiguation with information from the complementary channel. An added benefit of allowing speech and gesture for rotations is the ability for participants to communicate the degrees of rotation, allowing for more accurate interactions.

In the speech condition participants preferred to use <action> <direction> or <action> <object> <direction> syntax over complete sentences. Implying that both unimodal and multimodal speech utterances are syntactically simplified compared to conversational speech [113]. This is seen as saying "move" and "finger flicking" in the direction of the intended movement. In either case, the intended <action> was present indicating that full natural language processing may not be necessary for basic multimodal interactions.

This work contributes to findings on multimodal interactions and touches on some of the potential pitfalls of referent display which would cause reproduction to be difficult, as mentioned by

Villarreal-Narvaez et al. [48]. The impact of referent display on proposals is seen most saliently in the low $\mathcal{AR}$ for the *select* referent which often received high $\mathcal{AR}$ in prior studies [75, 99]. The timing information and patterns here provide insight into the formation of these interactions and extends the timing windows constructed by Lee et al. [9] by adding the phase of the interaction by the time of that phases initiation. This study gathers proposals within each modality allowing for comparison against gesture only studies [99], while also contributing to the less common multimodal elicitation literature [43, 51].

2.4.7 Design Guidelines

Instead of directly proposing a single set of consensus interactions within each modality we have chosen to show the distribution of interactions. By looking at these distributions a picture of trends across the top few proposals can be seen. For some referents, such as the translational referents, the top gesture in the gesture and the gesture and speech block matched (Figure 2.7). For translations often the top proposal was a reversible swiping gesture for moving the object in the x-axis and y-axis and an index extended swipe for movement on the z-axis. The speech proposals in these cases were also reversible (Figure 2.11). The first choice in all translations except *move down* was to say "move" and then a direction. For *move down* people commonly said "drop". *Create* and *destroy* followed the same pattern with a reversible bloom gesture either starting closed then opening or starting open then closing and the utterances "appear", and "disappear". For most gestures, a bi-manual version that was a symmetric two-handed version of the uni-manual proposal was also proposed (i.e. pushing with one open hand versus pushing with two).

Most gestures were based on the participants' understanding of naive physics, meaning how they perceived an object would react to an interaction as it would in the real world. Most variations occurred within specific hand poses but not the larger movements of the hand/arm. As such we recommend aliasing manipulative gestures across hand positions (open hand, pinch, grab) based on the type of movement. A second consideration should be made on the inclusion of bi-manual

gestures. while not found in abundance here, other work [43, 99] has found evidence that users may gravitate towards using them in other domains and with larger objects [73, 74].

Other referents had less consistency. In the case of *shrink* and *enlarge* a "bloom" gesture and two handed "pinch and drag" gestures were common. In this case, we would suggest reserving the "bloom" gesture for *create / delete* and allowing "grab and pull" and scaling as seen both here and in earlier work [99]. The top speech proposals for scaling were more agreed upon and should be implemented as well. Those were the reversible pair "enlarge", and "shrink". Rotational referents other than *roll clockwise* have high levels of disagreement among proposals. "spin" and "flip" should be enabled as action selection words then a gesture should be allowed for controlling the direction of the rotation.

Direct manipulations should be allowed when possible, especially for basic manipulations. Speech and gesture as multimodal interactions showed promise in areas where one or the other input lacked and should be included. Implementing a system such that it has an internal model of functionality that aligns with what most participants formed as their mental model of functionality would increase the user's chances of guessing the inputs. This would be most easily achieved with direct manipulations, which in this study were often very close to their real-world corollary.

Participants seldom used full sentences or referred to the object being manipulated (Table 2.10). Due to that word spotting should be sufficient for most tasks. Only two participants used full sentences and those sentences followed the <action> <object> <direction> syntax with prepositional terms added (e.g. "move to the right" compared to "move right"). In either command, the actual information content is held in the <action> <direction> terms which could be spotted. The use of simple commands when possible was also observed by [9].

The windows built around co-occurring interactions are incredibly useful to systems needing to decipher interactions. With segmenting interactions based on the first movement of a gesture, the transition into the stroke phase, the information content of both the speech and the gesture portions of the interaction can be found. In this study gestures nearly always preceded speech (405/408 proposals). Most commonly speech was around 81.67 milliseconds after a gesture initiated. The

stroke was often 90.87 milliseconds after the start of a gesture. Both of those phases represent the initiation of the actual information content of the interaction. The back end of these interactions is slightly less concrete. Often the end of a gesture preceded the end of an utterance. A system could be designed to use a time-out window after which the speech would be considered a separate interaction.

2.4.8 Limitations of the Study

By choosing to show animations for referents the gesture proposed may be biased to follow the animations shown. This choice was made to preserve the value of the speech proposals with pilot studies that showed speech was less impacted when showing the animations of the referents as opposed to the text. This study only allowed one proposal per referent per block. Having participants propose more than one interaction may have generated interactions that they felt more well suited to the referents. This study only showed a single virtual object at a time, which would impact the selection phase of any interaction. To help compensate for this we used the referent *select* independently.

For the rotational referents participants would sometimes use misaligned gestures and speech. They might say "roll clockwise" and perform a counterclockwise movement with their hand. Multimodal systems can suffer from compounding errors caused by incorrect recognition, or mismatched interactions such as the ones seen in this study [114]. These errors could take more time than standard uni-modal errors to correct or cause compounding errors when a second error is made during an attempt to correct the first.

2.4.9 Conclusions and Future Work

Several questions remain unanswered. If there were more than one object shown the gesture results would show more selection gestures. The choice of an object used could also impact the production of interactions. If a larger object or a differently shaped object was presented the hand postures used may differ. Future work should involve testing the proposals found here against ones produced by text-based referents to assess the impact of referent display.

Compound errors in uni-modal text entry systems cause a generally linear increase in correction time [115]. Recent work has shown that improved error correction methods can reduce the time it takes users to reconcile text entry errors, decreasing the overall amount the user is slowed down by the error correction process [116]. Further work is needed to examine whether this holds true for multimodal interactions.

This work presents a within-subjects elicitation study across three input modalities (gestures, speech, and co-occurring gesture and speech). By examining each modality independently direct comparisons between the changes in speech and gesture from unimodal interactions to multimodal interactions are shown. Trends in gesture proposals are shown at a granular level. Highlighting that while there is often disagreement in proposals given, that disagreement manifests as variations in with similar underlying formations. In gestures, this was a variation of the hand position and not in the gross movement. In speech, this disagreement is seen as consistency in the <direction> phrases used and minor variations in the <action> phrase (e.g. "move" to "go"). While a singular mapping of the top proposals would yield a consensus set that is discoverable to most users, by aliasing and understanding the likely variations in interactions, a larger percentage of users' natural interaction preferences can be captured.

This work extends the work of linguists [53–55], and the work of computer scientists [59–61, 95] into AR-HMD building environments. Timing windows based on the phases of co-occurring gesture and speech interactions as described by McNeil [53] have been constructed. Showing that in HCI the gesture stroke is closely aligned with the information content of both the gesture and the utterance given. These windows can be used to construct more accurate multimodal fusion models.

2.5 New Research Direction

These two studies generated findings on the types of interactions and behaviors people use while doing basic object manipulations in virtual environments when wearing AR-HMDs. While these multimodal interaction techniques show promise for simple tasks [29, 30], stereoscopic dis-

plays will likely be used to render complex environments, necessitating further research on user interactions in complex environments. IA was chosen as the complex environment to use for those examinations.

There are still many unexplored areas in IA and in multimodal interaction in AR-HMDs. Direct manipulations in IA may not vary much from generic AR environments [117] but other interactions such as panning, zooming, and more complex tasks like color selection or annotation techniques likely won't transfer as well. Additionally, the differences in interaction between AR and VR HMDs are largely unknown, making determining how interaction techniques could transfer from VR environments to AR ones difficult.

Before running any studies examining those interactions, a complex IA system needed to be developed. This system needed to be designed such that it could allow observation of interactions in complex environments utilizing both AR and VR HMDs and be used by more than one person at a time. Allowing multiple people to use it allows WoZ and collaborative interaction studies to be run in it.

Importantly, before an examination of multimodal inputs for complex environments could be run, the developed environment needed to be tested. There have been few works examining interactions in IA. Before interaction techniques can be developed for those complex environments, we need to understand what sorts of interactions will be performed inside of them and how users will navigate them. Apart from that, most work in IA has been conducted using VR-HMDs [11]. In order to leverage prior work done using VR-HMDs while developing for AR-HMDs, an understanding of how interactions differ between those two types of stereoscopic displays must be established.

The next chapter of this dissertation discusses the design and development of a complex IA platform that is able to run on both AR and VR HMDs. This system can be used collaboratively across these devices. The system provides tools that would be common to a data visualization platform such as highlights and text markup. This system is one of the first of its kind and as such stands as a contribution of this work.

Chapter 3

System Design and Design Choices

This system was developed over 3 academic semesters using a combination of expert-driven design and iterative design. This project was developed to a largely functional but unpolished state and then iterated on over 8 iterative design sessions and 5 pilot studies. The first 3 design sessions were informal usability sessions where participants were asked to interact with the environment, trying out each annotation tool provided while using a think out load protocol. Using this protocol, participants were asked to comment on what they were trying to do, what was working, what was not working, and what would improve the system.

The other 5 sessions occurred once the platform was close to completion. These later sessions gathered user system feedback and interaction preferences. The last 5 pilot sessions were used to fine-tune the full environment and to test the experimental tasks that would be used in an AR VR interaction comparison study (Chapter 4). The design sessions and pilot studies are elaborated on further in the next chapter that details the AR-VR comparison study. They are mentioned here to provide context around the design process and how/why design choices were made.

3.1 System Overview

This is a multi-user cross-device IA system with annotation and marking tools. This system is able to load a scatter-plot visualization from a .csv file and render it in 3D stereoscopic environments. That scatter-plot can be marked up and annotated using tools provided in the system. Multiple people can log into the system at the same time, allowing them to see and interact with the same content. If these users are co-located, the virtual content can be synchronized between their devices allowing local collaboration. If these users are remote, their presence is represented virtually so that other connected users can see where they are standing/pointing/interacting. A key feature unique to this environment is its ability to run on a range of devices, including AR-HMDs, VR-HMDs, and a computer.

This is one of the first systems of its kind, and as such the iterative design sessions helped to refine many of the decisions made during its development. This chapter covers the key design choices that came from these iterations and details the final design used for the AR VR comparison study. This chapter can help other researchers to develop complex AR/VR environments and help readers to better understand how this platform behaves.

3.2 Interactions

The main interaction technique used in this system is ray-casting. Ray-casting is an interaction technique where a line with a cursor at its terminal end is projected forward from a device. On the AR-HMD, a ray-cast is projected from the user's hand and selection is triggered by pinching their index to their thumb (Figure 3.1). In VR, ray-casts are projected from the VR-HMDs controllers and a button push is used for selection (Figure 3.2). When using the ray-cast, a dotted line is extended from the user's hand/controller. This line is labeled "Ray-cast" in figures 3.2 and 3.1. At the end of that line is a cursor that looks like a white open circle. When the selection interaction is used (e.g., pinch in AR, button press in VR) the ray-cast line becomes solid and the cursor turns into a filled-in white circle, labeled with "(selected)" in figures 3.2 and 3.1. In addition to the ray-cast, participants in the AR group were able to push buttons with their index fingers.

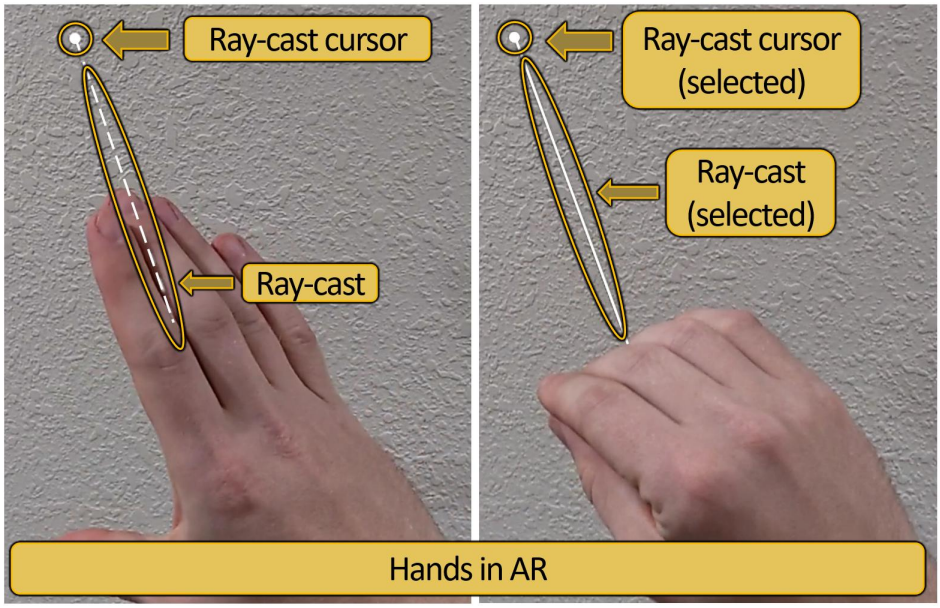


Figure 3.1: Labeled hand selected and regular ray-casts. Ray-casts are enhanced due to low AR capture resolution.

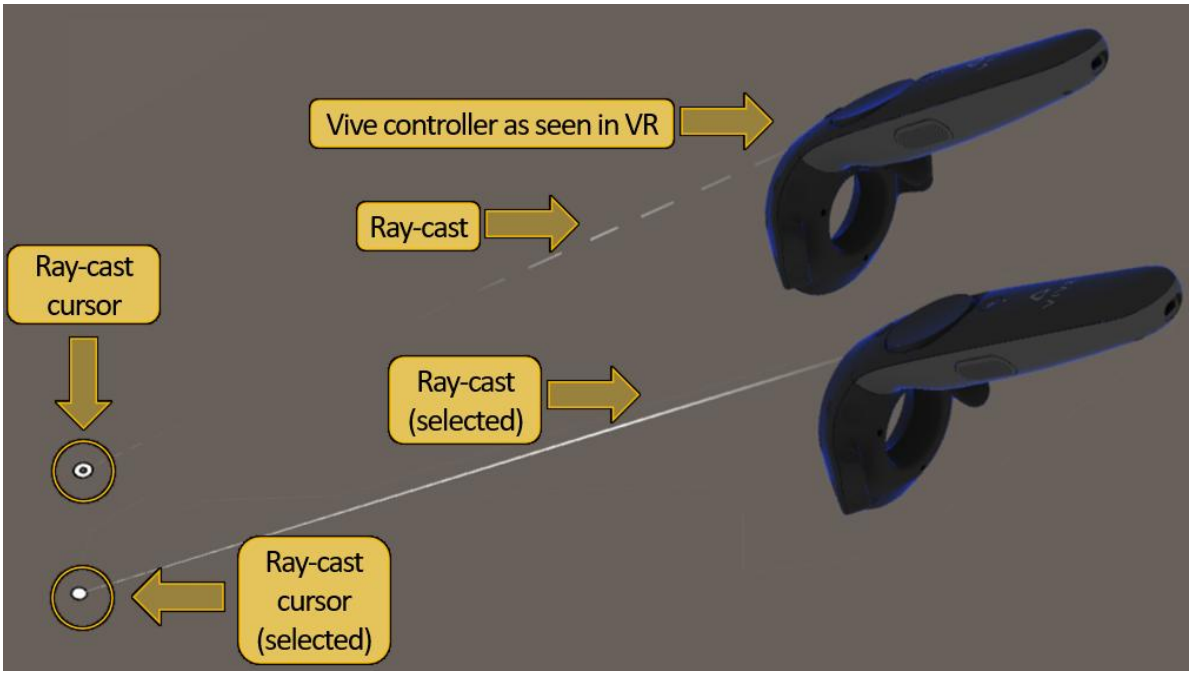


Figure 3.2: Labeled Vive controller selected and regular ray-casts.

3.3 Visualization

This section covers the design choices that went into displaying and interacting with the scatter-plot visualization used.

3.3.1 Data-set

This system is able to load .csv format data sets containing strings and/or numbers. This research used a dataset of common cereal's nutrition facts, chosen because it provides an accessible topic and has dimensions that are well displayed using a scatter-plot. The dataset includes carbohydrate, sugar, protein, fat, and dietary fiber content for each individual cereal. The cereal names were included in this dataset but were excluded from this task. The manufactures of the cereals were used instead, providing a smaller space of values for users to remember. The wine quality dataset was considered; however, the dimensions included in the dataset were less approachable by users unfamiliar with wine (i.e., tannin, sulfates). The .csv reading and loading base code was provided by the Immersive analytics toolkit (IATK) [1]. The cereals data-set used by this work is provided in Appendix A.5.

3.3.2 Scatter-plot Visualization

The earliest version of this system utilized an immersive analytics platform called DXR [118]. DXR was less compatible with the MRTK necessitating a switch to the IATK [1]. Note that both DXR and IATK were developed to use SteamVR which runs on VR-HMDs. The IATK was heavily adapted to allow it to run using the MRTK in a multi-user setup, allowing for AR-VR cross-device use. The MRTK and SteamVR are both built on OpenVR, a C++ library for VR development. This library is wrapped so that it can be used by Unity's C# scripts. The IATK provided the ability to generate and view the scatter-plot visualization.

There are several visualization options provided by the IATK including a bar-graph, trails and trajectories, scatter-plot, and connected dots [1]. The scatter plot was chosen because it provided the best means of seeing values in the cereals dataset and provided a nice base for data annotation

tools. A scatter-plot showing fat, dietary fiber, and sugar content is shown in Figure 3.3. The same scatter-plot with its points size and color mapped to fat content is shown in Figure 3.4. In Figure 3.4, a ray-cast intersection point can be seen in the lower left causing light pink colored visual proximity feedback and the appearance of the scale and rotation handles.

The only features used from the IATK were the loading of .csv's, the base rendering for the scatter-plot, and the programming interface for dimension mapping changes [1].

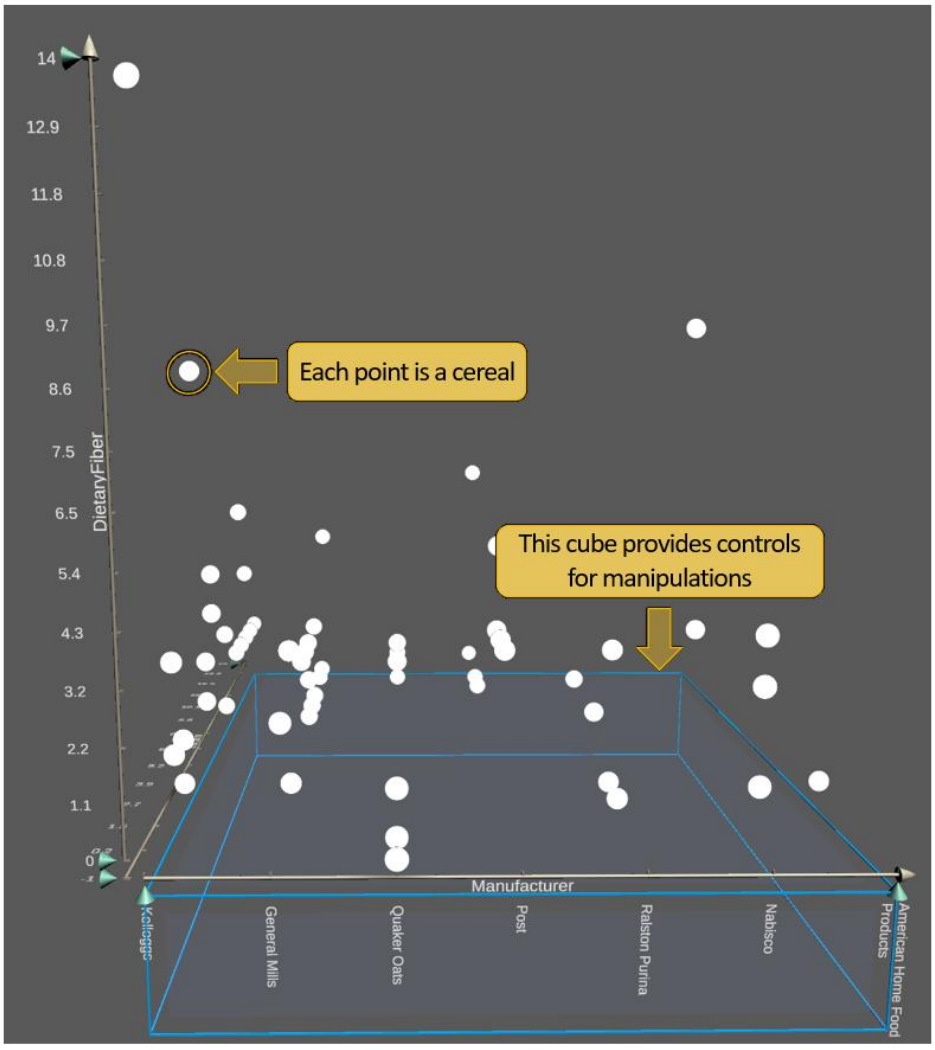


Figure 3.3: Labeled scatter-plot visualization, the manipulation controls cube provides interaction controls when hovered over.

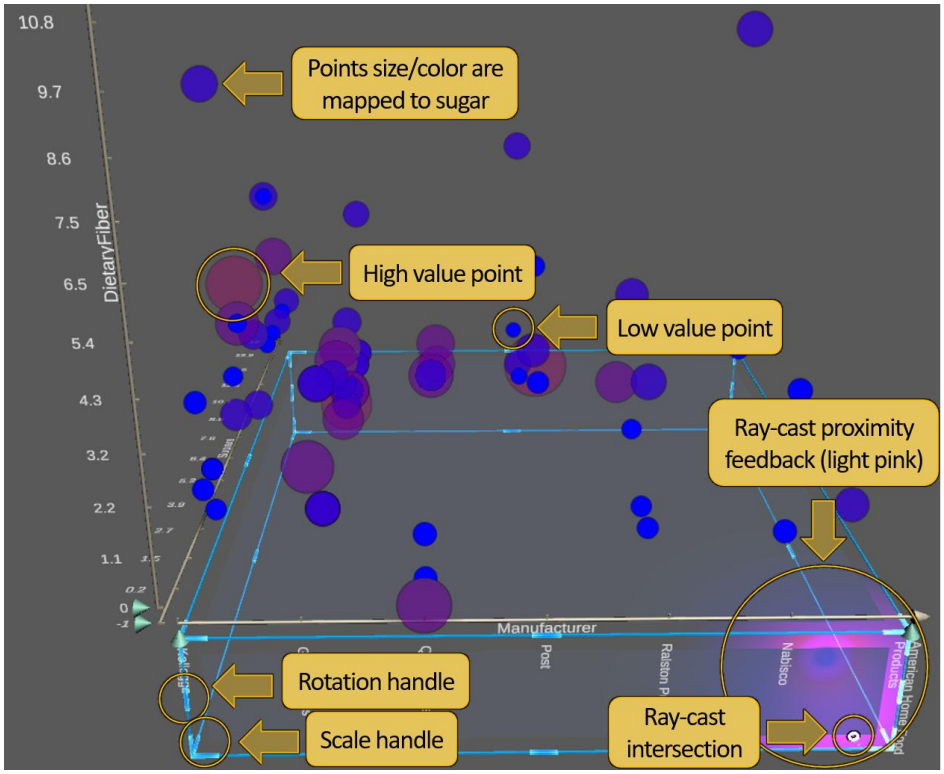


Figure 3.4: Labeled scatter-plot visualization with a color/size mapping that has a ray-cast intersection on the lower left.

3.3.3 Visualization Interface

The original visualization interface consisted of 5 drop-down menus that were shown in a single view pane (Figure 3.5). These drop-down menus would populate options based on the dimensions in the CSV provided. They would disallow any axis (i.e., drop-down menu) to be equal to each other but would allow any dimension to be mapped to any axis. Early iterative design sessions showed that the original drop downs were too difficult to interact within AR. In VR they were manageable but were not considered easy to use. These usage difficulties came from the number of precise ray-cast interactions needed to navigate the menu. Users had to select the drop-down, then select the opened drop-down menu and drag it down to scroll to the intended item, and lastly, select the opened item.

The first approach to improving the drop-down menus was to reduce the number of interactions required of participants to select an axis. By only providing three-axis options per drop down, all options could be displayed at once, thus no longer requiring participants to scroll through the list.

Figure 3.5: Early visualization interface utilizing drop-down menus

This adjustment only slightly improved users' interactions with the menu as they still struggled to select the dropdowns to open the three-option menu.

Due to the difficulty of selection and the general need to streamline interactions, a new system was designed where participants could use buttons to switch between axis and size/color mappings (Figure 3.6). In this design, there are three rows of buttons to adjust axes with each row labeled with the axis and the currently mapped value. Adjacent to but independent of those buttons were 4 rows of buttons that change the size/color mapping with an option for "undefined" which removes the color/size mapping. Size and color were now mapped to the same value instead of having the option to set each independently, further reducing the complexity of the button system. The number of axes mapping options was set at two to streamline the experience of the user. With the button system, participants were much more likely to successfully change an axis mapping and were more likely to switch between axis mappings.

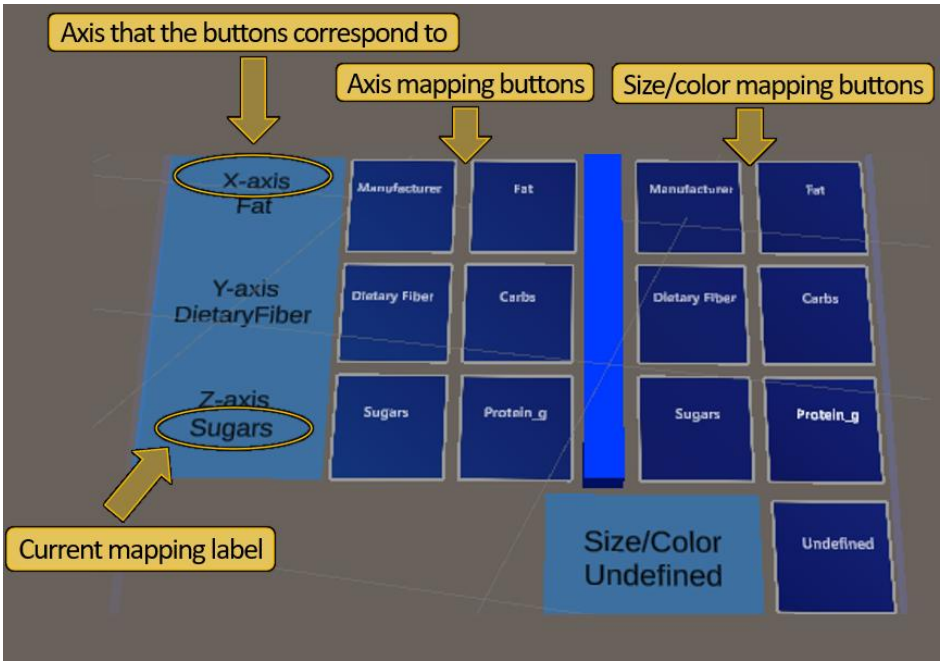


Figure 3.6: Labeled final visualization interface utilizing buttons. The current mappings are shown in text beside the buttons that would change that mapping.

3.4 Annotations

A prominent component of this system are the provided annotation tools. There are 6 types of annotations: details on demand, mean/median plane, text/speech input, sphere highlight, cube highlight, and a drawn mid-air line. These annotations have been built up over time each going through several iterations. All annotations can be placed, generated, and manipulated by any user of the platform (e.g., the system is collaborative). When an axis of the visualization is changed, the annotations currently displayed are removed and any annotations that were generated for the new axes combination are loaded allowing participants to return to their previous annotations when exploring the dataset. As an example, annotations made when the visualization displayed fat, carbs, and protein are removed when the visualization displayed fat, carbs, and sugars. When the first set of axis mappings is returned to the annotations made for it are loaded. Annotations are saved independently of the size/color mapping.

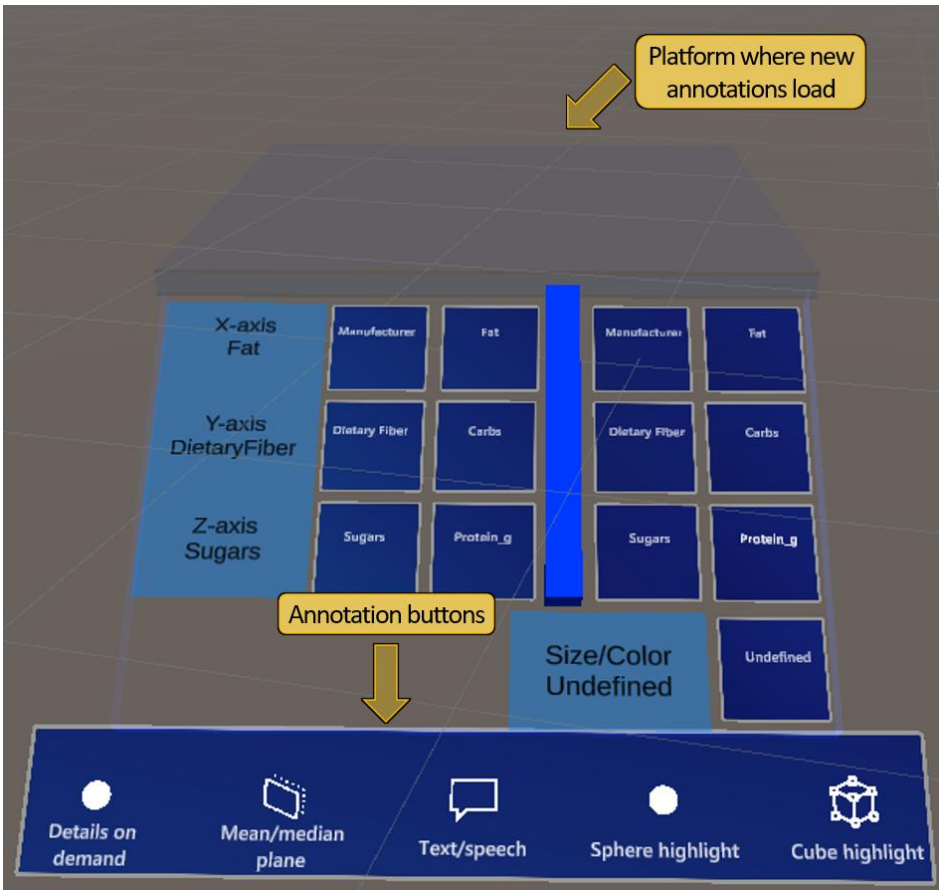


Figure 3.7: Labeled final visualization interface and annotation controls. The platform holds spawned annotations where the button controls interface with the visualization or call for annotations to be made.

3.4.1 Generating Annotations

Early versions of this system generated annotations by using a small menu pane of annotation tools that could be set to follow the user. The "following" function confused users who expected the menu to stay stationary. A separate issue was that of where to spawn annotations. Early versions of this platform loaded new annotations above the visualization and beside the visualization; however, both of those locations were difficult for new users to locate. Users often had to scan the environment for new annotations, slowing down their interactions with the visualization. This was resolved by building an annotation control system into the visualization interface (Figure 3.6). A platform was added above the visualization interface and the annotation spawning functions were taken from the user's "following" annotation menu and added below the visualization interface.

The final design can be seen in Figure 3.7. With this design, annotations consistently spawned in a location that was near where the buttons for loading them were placed.

IA environments are visually rich and viewing data in stereoscopic 3D is unfamiliar to most users. Therefore, limiting the number of interactions new users need to initially learn before using the system is critical. Reducing the required learning effort will improve a new user's experience in the environment and help mitigate feelings of being overwhelmed by the environment.

3.4.2 Deleting Annotations

Early versions of the system used a button attached to each annotation and a virtual delete tool. One button per annotation introduced too much visual clutter where having a separate tool to learn how to activate and use increased how much a new user needed to learn. The final solution used was to allow annotations to be deleted by dragging them to a trash bin (Figure 3.8) that was loaded to the right of the visualization and can be placed anywhere by the user. This tool is labeled "Move annotations into here to delete them" providing users with a clear message of what the tool is and how to use it.

3.4.3 Details on Demand Annotation

The details on demand (DoD) annotation generates a small sphere that can be moved around the scatter-plot (Figure 3.9)indicating the x,y, and z-axis values for the DoD sphere's current position in the data. Moving the DoD sphere around the scatter-plot allows users to observe what the values are for any position in the scatter-plot. At the scatter-plot point nearest to the DoD sphere, a second sphere that is transparent with a light pink outer ring appears ("Nearest point value" label in Figure 3.9). This sphere displays the x, y, and z dimensions of the real data-point (e.g., a data-point represented by the scatter-plot) nearest the DoD sphere. An additional display is given on the axis in the form of text labeled bars ("Axis ticks for nearest point value" label in Figure 3.9). These bars were first mapped to the move-able sphere's position to provide participants a way to mark ticks on the axis. They were later changed to match the closest real data point's position to assist participants with ease of data interpretation. Regardless of which sphere's data the ticks are

Figure 3.8: The "trash bin" annotation deletion tool, activated by moving an annotation to it.

mapped to, they provide a visual x, y, z mark that helps indicate the placement of the selected point in 3D space. This visual aid is beneficial when viewing centrally located data points where there are few visual cues to the actual location of a selected point.

3.4.4 Highlight Volume Annotations

This system provides both cube and sphere-shaped highlight volume tools designed to allow both non-uniform and uniform scaling (Figure 3.10). Using the ray-cast to grab the middle of a side, as aided by a visual handle icon, allows rotation about a single axis. The first version of this platform enabled highlighting of points using a shader, a visual texture change that happens when rendering what the user sees. The graphics rendering script (shader) would determine if the highlight volume was above a rendered point in the scatter-plot to change the color of that point to

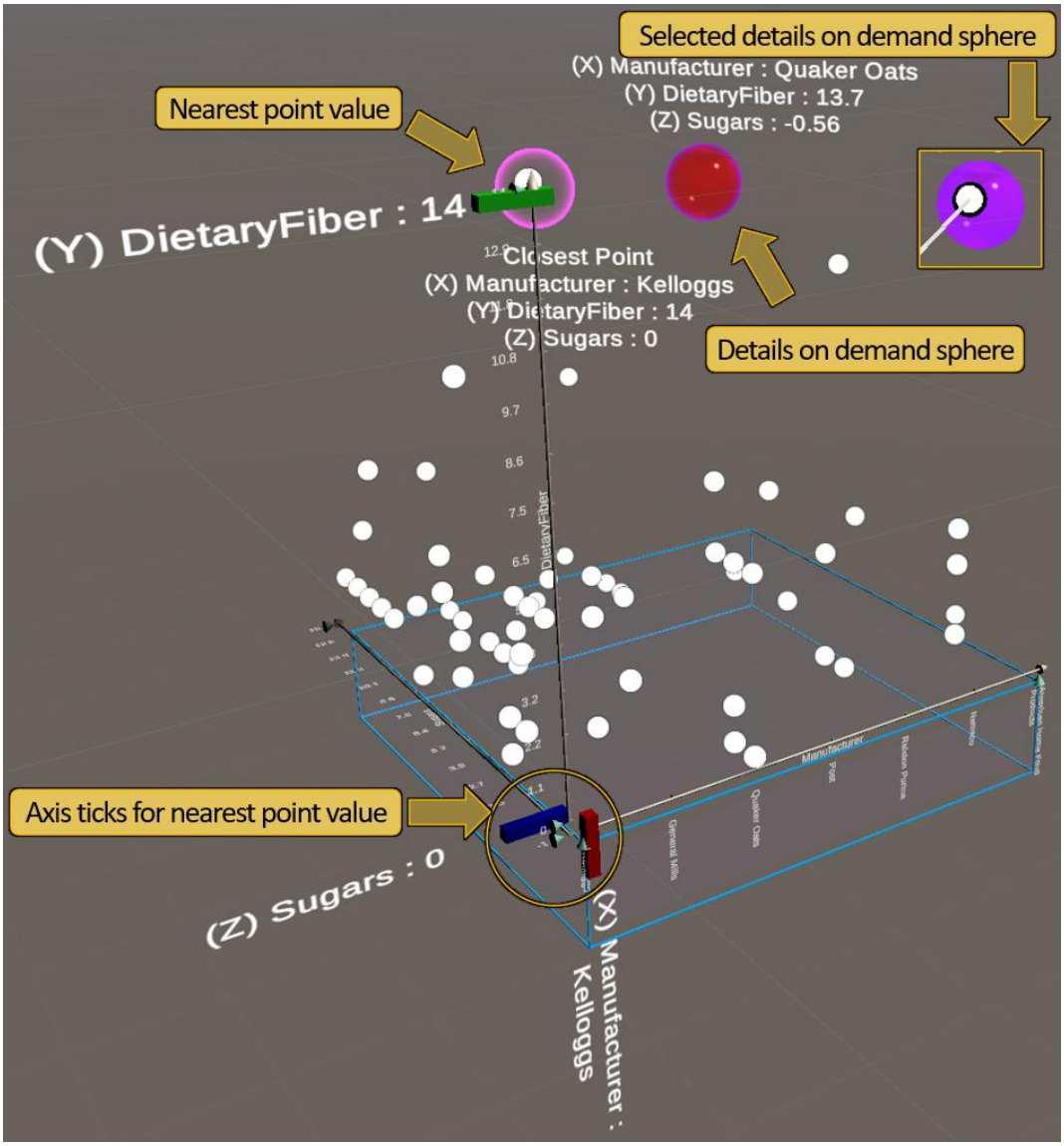


Figure 3.9: Labeled scene showing the details on demand annotation

red. This approach was not compatible with HL2 due to a difference in shader pipelines between the HL2 and the VR-HMD.

The next iteration generated red spheres and positioned them over existing data points. This solution worked on all platforms but required more objects to exist in the scene, increasing the computational overhead of the annotation. These spheres are generated on each client locally based on the position of the highlight volume which reduced the network traffic needed to synchronize highlights between users.

Highlights include a text label indicating the number of data points encapsulated within the highlight (Figure 3.10). This feature was included after several participants in the pilot studies noted difficulty counting data points in 3D space. Minimum and maximum scale constraints were also added to compensate for the sometimes difficult-to-use non-uniform scale interaction. Under these constraints, users could not reduce the scale of any axis of the object below 33% of the original size or above 150% of the original size. These constraints were deliberately large to prevent users from shrinking objects to a point where they were no longer selectable due to occlusion caused by the rotation and scale handles.

3.4.5 Text Annotation

The text annotation tool consists of a handle, a text input box, and a dictation button (Figure 3.11). Clicking on the input box displays a virtual keyboard unless a Bluetooth keyboard is connected, in which case the Bluetooth keyboard is used. Pressing the dictation button initiates the recording and processing of speech for text entry. A manipulation handle can then be used to move the annotation around the scatter plot. Both the dictation button and the manipulation handle use minimalist backgrounds until hovered over to minimize visual clutter when used on the scatter-plot (Bottom of Figure 3.11). The text input box automatically resizes to fit the size of the text contained, seen when comparing the top text box to the bottom one in Figure 3.11.

3.4.6 Centrality Annotations

A centrality metric plane that attaches to a plane representing the mean or median of one axis on the scatter-plot was also implemented as an annotation tool (Figure 3.12). The planes are colored red, yellow, and blue for the X, Y, and Z axes respectively. This tool featured 5 buttons. These buttons could be used to change between viewing the mean and the median of a given axis and to set the axis that the tool was attached to. When the user's ray-cast cursor hovers over one of these buttons, they expand to indicate that they are interactive. This expansion can be seen when comparing the "Mean" button to the "Median" button of the z-axis (blue) plane shown in Figure 3.12. The button area gains a semi-transparent background when hovered over to show

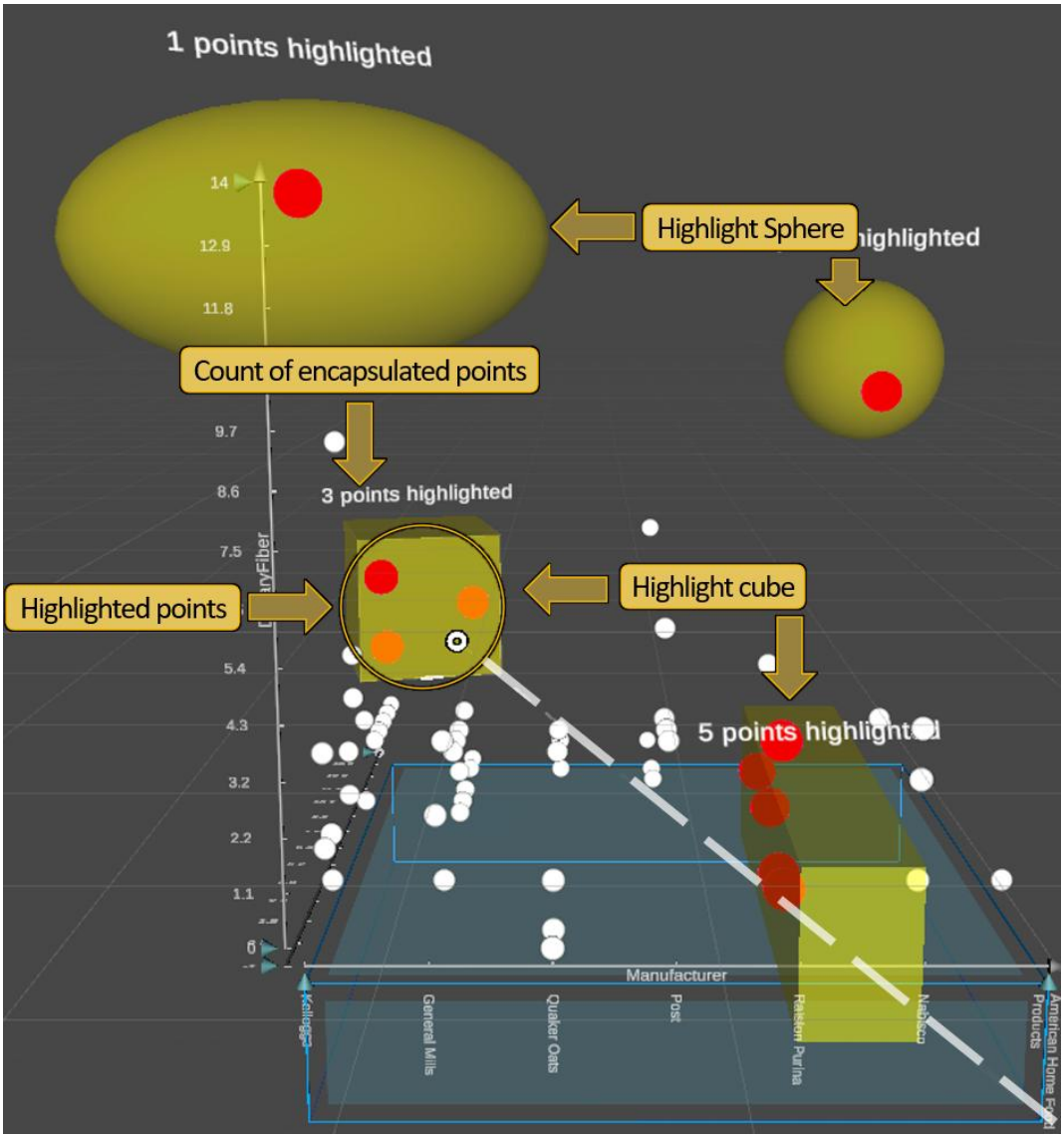


Figure 3.10: Labeled scene showing the cube/sphere highlight volumes and a ray-cast with an open cursor.

where the plane can be grabbed. A text indicator displays the value that the plane is mapped to. During the iterative design process, this tool was improved by attaching the buttons to a separate panel that faced the users head position regardless of the centrality plane's orientation. This ensured that the buttons and text were always visible to the user. The larger colored plane has no physics allowing ray-casts to go through it. This way users can interact with content behind the plane.

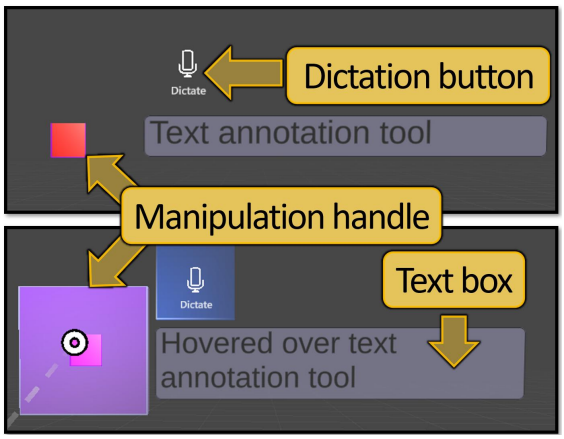


Figure 3.11: Labeled scene showing the text annotation and a ray-cast line+cursor

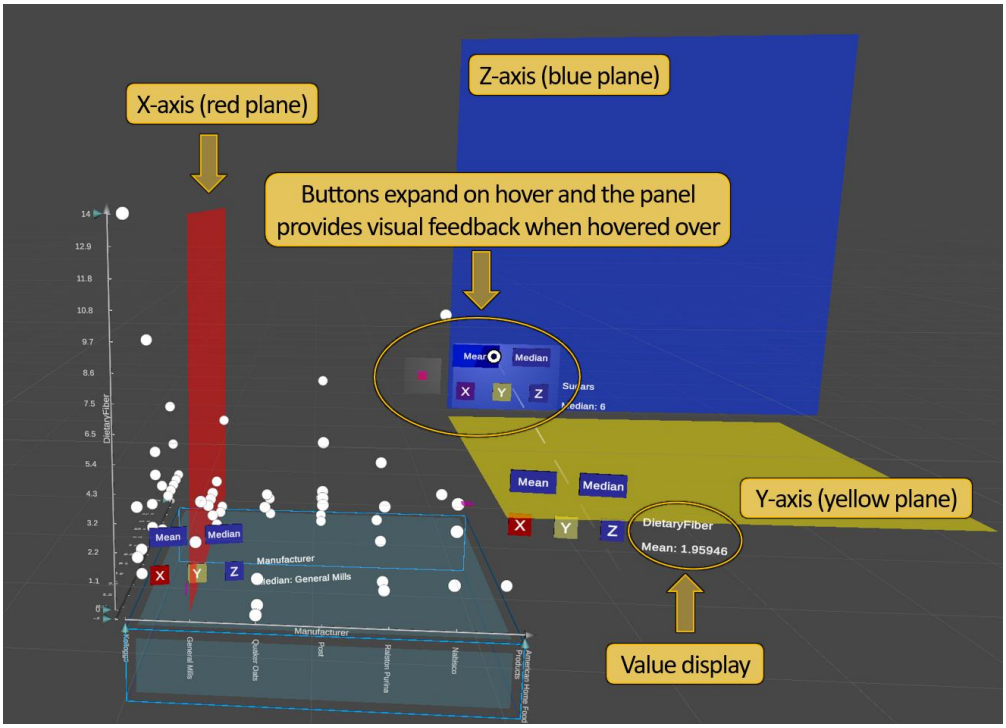


Figure 3.12: Labeled scene showing the mean/median plane annotations. The y and z axis planes have been moved off of the visualization for this figure.

3.4.7 Mid-air Line Annotation

Users are able to draw mid-air lines using a Logitech VR Ink [8] mid-air 6 degrees of freedom (DoF) pen input (Figure 3.13). By having 6DoF, the pen's movement and rotation can be tracked on the x, y, and z axes.

[8]https://www.logitech.com/en-us/promo/vr-ink.html

To start drawing a line, users could either press the tip of the pen to a surface or press a button near the front of the pen (Figure 3.13). Once either of these conditions was not met the drawn line was finalized. These lines can be drawn anywhere in the environment.

After being drawn, lines could be manipulated by users (i.e., rotated, scaled). Allowing these manipulations originally required the drawn line to be encapsulated in an invisible cube, referred to as a bounding cube, that provides the rotation and scale handles. These bounding cubes could potentially obstruct large areas of the environment, making them inaccessible for ray-cast selection. However, removal of the bounding cube required users to accurately select the line itself increasing the user difficulty.

The next version of this tool disabled the bounding cube after 15 seconds of the line being finalized under the theory that users would immediately position the line after drawing it. This version did not allow users to remove or adjust the lines after 15 seconds had passed. During the iterative sessions, participants occasionally needed to delete lines when revisiting previously marked visualization states, making this approach sub-optimal.

The final solution was to remove the bounding cube 5 seconds after line completion and to allow users to re-enable the bounding cube by passing their ray-cast through the drawn line. While this still requires users to intersect the line with their ray-cast, they no longer needed to use selection commands on the line itself (i.e. pinching of thumb and index finger). By removing the precision needed during the selection of drawn lines the handshake associated with using selection commands was no longer an issue.

During development, the line thickness was decreased from 2.3 millimeters to 0.0635 millimeters. This was changed after three of the pilot study participants noted that it would be more useful if the mid-air pen's line was similar to a normal pen's drawn line. The last improvement to the line was to automatically remove line segments that are shorter than 12.7 millimeters. This removed most of the lines drawn by accident when re-positioning the pen in the user's hand or when putting down the pen.

When users in the AR-HMD held the pen, their hand's grip around the pen would inconsistently cause their ray-cast to select because of the distance between their thumb and index finger. This caused AR users to unintentionally grab and move objects when drawing with the pen. This was resolved by disabling a AR-HMD user's ray-casts when the grip button on the pen was pressed. The pen's grip buttons are shown in the lower right of Figure 3.13.

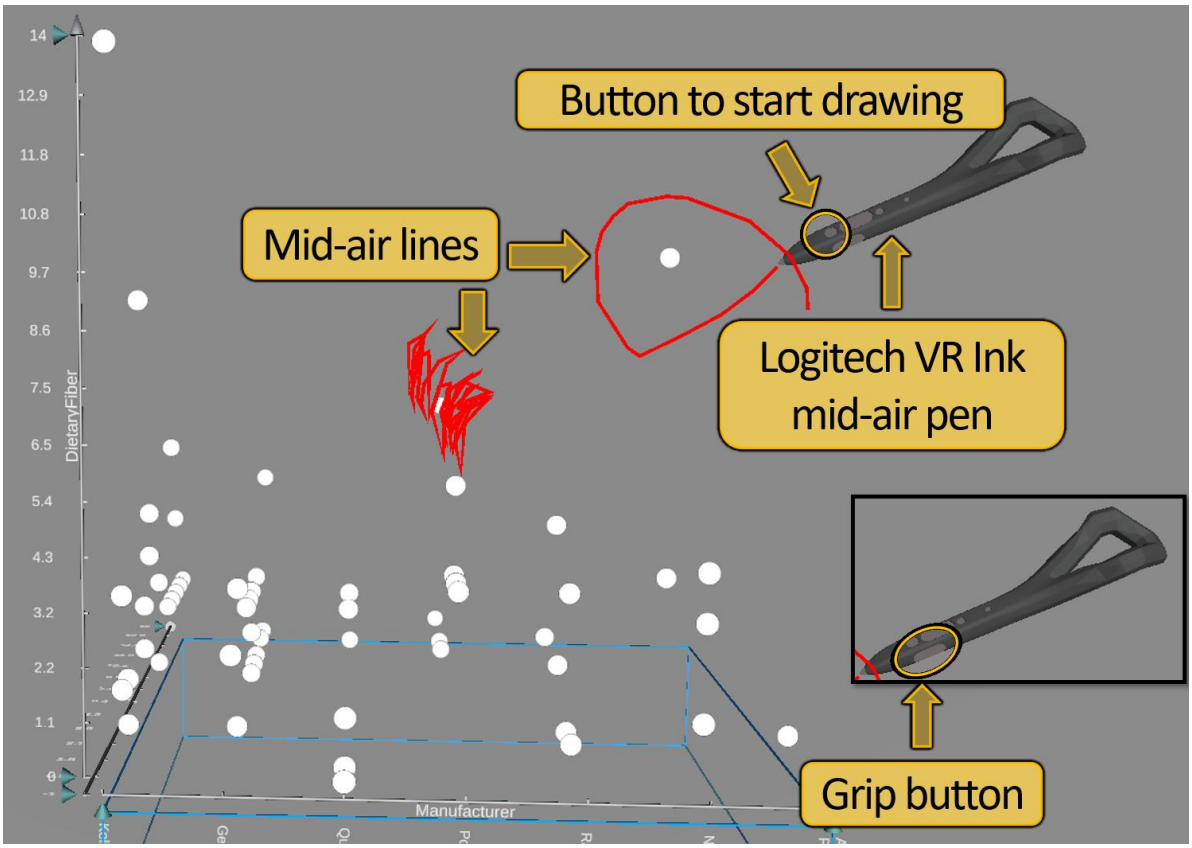


Figure 3.13: Labeled scene showing the mid-air pen model and two drawn lines.

3.5 Feedback Mechanisms

Over the iterative design process, many feedback mechanisms were added to aid users with their interactions in this system. These feedback mechanisms were critical to allowing participants to navigate this environment. Without them, participants could not tell what manipulations

were offered by different objects in the system or when their interactions with those objects were recognized.

3.5.1 Feedback for Object Translation, Rotation, and Scaling

When interacting with objects, both visual and audio feedback are used. Two of the types of visual feedback provided for manipulations are shown in Figure 3.14. The first column in Figure 3.14 describes the type of feedback detailed in that row of the figure. The second column shows the feedback for the visualization, which was one of the two main types of visual feedback provided to users. The last column in Figure 3.14 shows the visual feedback provided for highlight volume annotations, the other main type of visual feedback. The cube highlight annotations are shown over a sphere highlight to signal the transparency of the feedback states.

The first row of Figure 3.14 shows the base visual states of items that can be manipulated using handles. In an object's base state, the object appears as it normally would with no alterations to its appearance. The second row shows common hover states which are divided into two main feedback categories. The visualization displays the full bounding cube dimensions along with proximity-based visual feedback shown as brighter colors fading out from where the ray-cast is intersecting the bounding cube. Annotations use slightly different feedback where only the bounding cube's edges that are nearest to the ray-cast intersection are shown. This reduces some visual clutter from annotations inside of the visualization. Additionally, on annotations, the rendered edge of the bounding cube shows the proximity effect (right side of the second row in Figure 3.14).

The last three rows of Figure 3.14 show the visual feedback for translation, scaling, and rotation. Translation is initiated by selecting an object while the ray-cast is in contact with the object's bounding cube. Once an object is selected for translation its color changes to a semi-transparent red color (third row in Figure 3.14).

To trigger a scaling or rotation action, users must select handles that are shown along the edge of an object's bounding cube. On annotations, these handles start at a smaller size and are shown in white. As a ray-cast intersection gets closer to the handle, it expands and turns blue. When

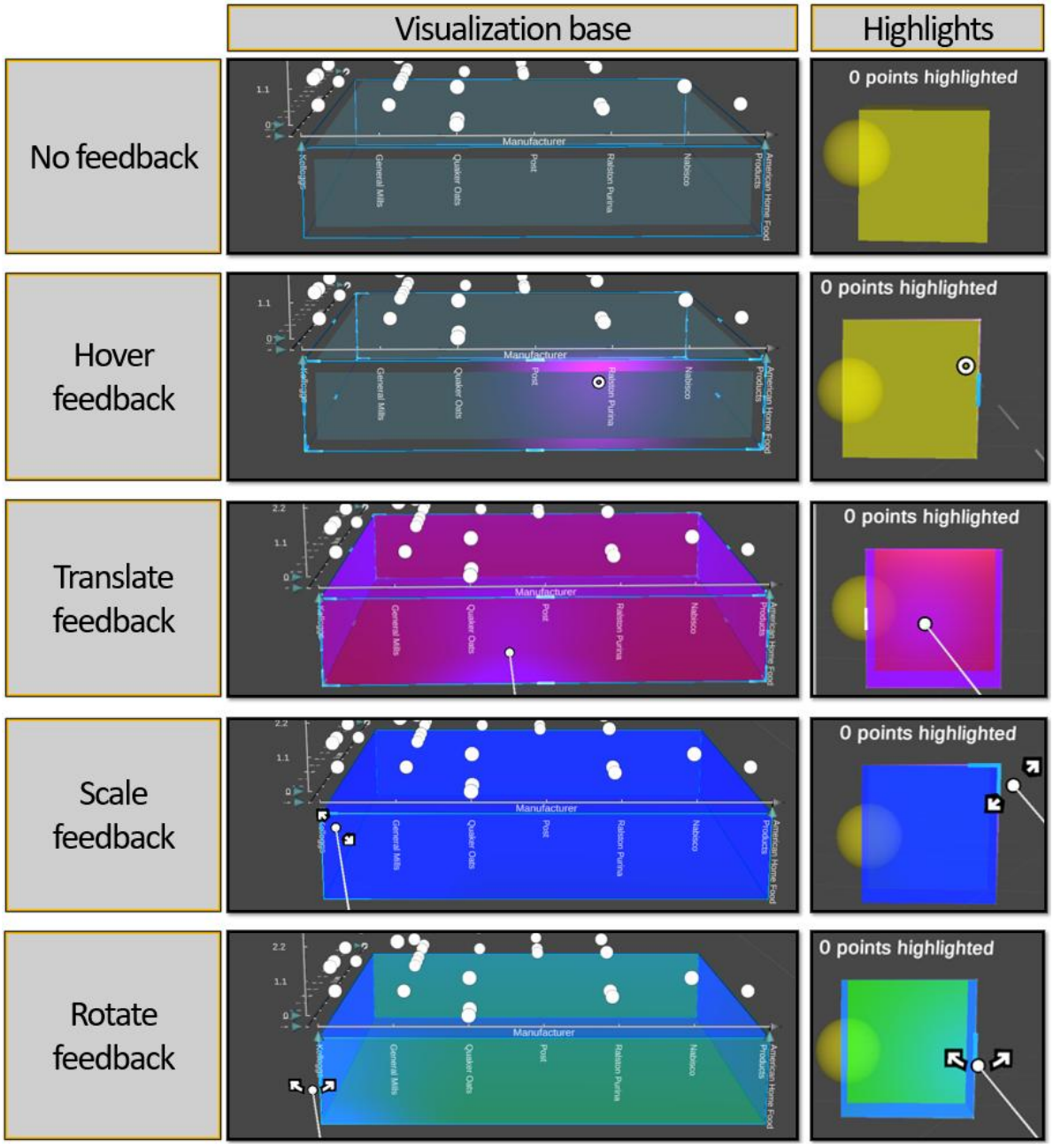


Figure 3.14: Visual feedback given for manipulable items base, hover, translate, rotate, and scale states.

a handle is hovered over the user's ray-cast cursor changes shape to a double arrow with either curved arrows for rotations or straight arrows for scaling. Once these handles are triggered the object's color changes to match the command given. For scaling the object is turned to a semi-transparent blue, and for rotation, the object becomes a semi-transparent green. The object's base

state and the color change for translation, rotation, and scaling are shown to all users such that if one user is translating an object all users in the system see that object turn red and moving.

In addition to visual feedback, audio feedback was provided for each type of manipulation (i.e. translation, rotation, scaling) on both selection and release of the object. When a manipulation is first triggered (i.e., a ray-cast selection is made) an approximately $1/5^{th}$ of a second long audio clip is played. When the selected object is released a similar audio clip with more bass emphasis is played. The audio clips used for each type of manipulation are notably different, where the selection and release clips for each manipulation type are similar. In total there are 6 audio clips for manipulations, one each for the selection and release of the object or manipulation handles.

3.5.2 Menu and Button Feedback

The basic buttons used in the menu systems were provided by the MRTK (Figure 3.15). The MRTK buttons were altered to show larger font. The base menu state, shown in the top left of Figure 3.15, consists of a blue background with text and image icons. Hovering over the menu causes button outlines to appear in white. When pressed, the buttons visually compress and play an audio clip. When selected, the menu turns light blue and can be moved by users. A thumbtack icon on the bottom right toggles whether or not the menu follows the user's position. These buttons were enhanced with visual proximity feedback where they become larger as a ray-cast line moves closer to them, shown in the buttons on a blue plane in Figure 3.12. When pressed these buttons play the same audio clip as the menu buttons.

3.5.3 General Translation Feedback

Many objects used in this environment have been designed to minimize opaque surfaces to enable users to visualize data behind their annotations. In addition to seeing beyond objects, users also need to be able to interact through objects. Imagine that the visualization is tilted away from the user with a mean plane set to the y-axis. If a user wishes to move a DoD annotation to the top of the visualization and the shortest path of travel is through the mean plane, the user should be

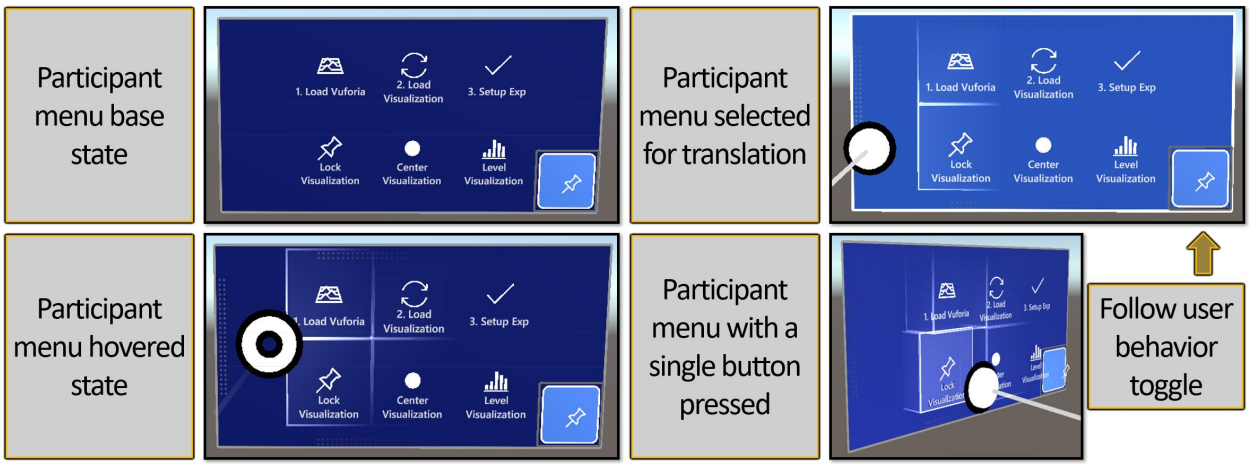


Figure 3.15: Participant menu with visual feedback for menu and button interactions

able to move the DoD sphere through the plane. For this to work, the mean plane cannot trigger a collision with the DoD sphere or the user's ray-cast.

Objects that do not need a handle are given either the full manipulation controls as seen with the highlight volumes and the visualization (Figure 3.14), or they are given a base material that shows proximity effects indicating the object can be interacted with (Figure 3.16). Translating these objects will trigger a color change and will play an audio clip on selection/release.

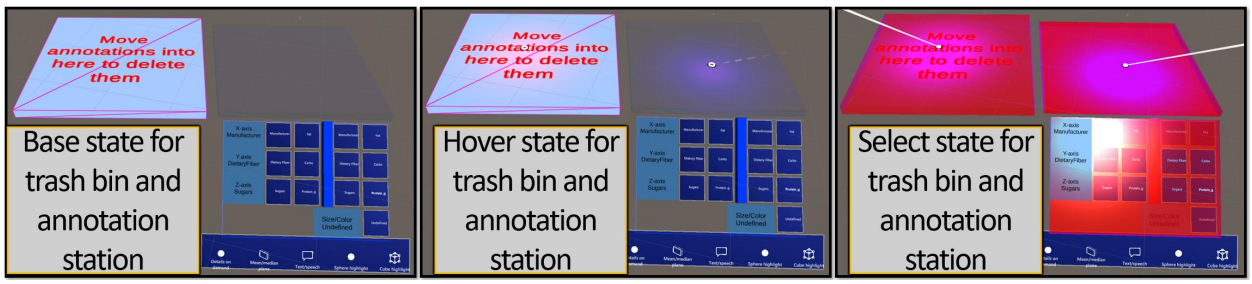


Figure 3.16: Feedback given for objects that do not have a bounding cube but can be translated.

3.6 User Experience Improvements

This section covers the improvements that were made to this system out of the iterative design sessions. These improvements were made to reduce the friction of executing commonly used sets of interactions.

3.6.1 Level Button

The first of these improvements was a button that levels the visualization in reference to the real-world ground (bottom right icon in the menus shown in Figure 3.15). After using the rotation handles, the visualization may require multiple interactions to return it to a relatively level state. The level button allowed users to feel more comfortable with rotating the visualization, with the knowledge that it could be easily leveled.

3.6.2 Center Button

A button was added that centers the visualization in front of the user (bottom center icon in Figure 3.15). At times the visualization can be moved to an inaccessible spot or a hard to recover from location. An example of this is seen when moving the visualization with the Vive controllers. If the controllers lose tracking mid-movement, the controller's virtual position might drift a few feet from the user causing the visualization to also move. The center button allows rapid retrieval of the visualization in those scenarios.

3.6.3 Lock Button

Any part of movable objects that intersect the user's ray-cast, including buttons, can be used to select and move the object. Grabbing a button and moving the ray-cast will move the object and cancel the button press. This design provides users with a larger surface area available to intersect when moving objects, reducing the precision needed for these interactions. The downside of this design is that when a user tries to press a button, if their controller or handshakes, the interaction may register as a move command rather than a button press. This was resolved with the addition of a button that locks the visualization and the annotation station in place by disabling the object's translation controls (bottom left icon in the menu shown in Figure 3.15). The visualization can still be rotated and scaled while locked. The lock allows the buttons on the annotation station to be pressed without any movement issues. It also helped prevent people from accidentally se-

lecting and moving the visualization when they were intending to move an annotation above the visualization.

3.6.4 Identification Entry

The first version of this platform used a text input box that followed the user for them to enter their assigned participant identifier (PID). This design was changed to allow any user to remotely change any other user's PID. The PIDs were then set to load from and save to a file allowing them to also be changed prior to a participant's session.

3.6.5 Setup Button

A number of setup steps must occur prior to a participant interacting with a visualization in this system. Most of these steps involve the removal of unnecessary visuals or objects from the environment. As an example, the representations of other players are removed from the participant's instance of the platform when they are not performing collaborative tasks. A setup button was added to minimize the number of steps a researcher or a participant needed to take before the experiment can start. This button is shown on the top right of the menus in Figure 3.15. The setup button removes any controls or objects that are not used during the experiment. It removes all representations of other players and places the visualization in the center of the image targets (if done after recognizing them). The image targets were three letter paper sized images that were printed and placed on the desk in front of users. These image targets are used to synchronize the environment across devices so that when collaborating in person, both people see the same placement of items. This process is further detailed in Appendix A.3.

3.7 Wizard of Oz Capabilities

This environment was developed with the goal of being able to use it to run Wizard of Oz (WoZ) style studies. In a WoZ study, a human acts to recognize the participant's inputs and triggers system responses accordingly. This allows researchers to test interactions that are not yet implemented.

In this system, the wizard has access to the same controls as the participant as well as a control menu that offers deeper access to all running sessions of the platform. The control menu can be used to adjust the coordinate synchronization location of another user. When the wizard uses the Unity editor instead of an AR or VR HMD, they may use the built-in translation, rotation, and scale tools that unity provides when viewing a scene. These controls offer a more precise means of interaction than the ray-casts. It should be noted that positioning 3D content using the editor can be more difficult due to the challenges of 3D navigation in a 2D environment. Even so, the Unity editor controls provide a more precise means of translating or rotating objects on a single axis.

3.7.1 Control Panel

A control panel menu is provided in the environment for the wizard or researcher to use (Figure 3.17). This panel can be disabled from loading in participant sessions. It can also be removed from the environment at run-time.

Figure 3.17: Extended control menu provided for the wizard and researcher.

Session management The wizard is able to set their own PID by pressing the top button in the first column of Figure 3.17. In the same row of buttons, they can load/remove the visualization, load the base level (e.g., the entry point), and set up an elicitation study. Loading the entry point removes any virtual content that is not necessary for the base experimental environment (e.g., a loaded image target). Pressing "Set up Elicitation" causes all other user's instances of this environment to remove any interaction tools deemed unnecessary for an elicitation study. These include object manipulation controls and menus.

The wizard is also able to force VR-HMDs to scan for new controllers by pressing the "Get Devices" button. This is beneficial in cases where a new input device is connected but not recognized as connected. The wizard can also hide or show controller models on all other clients by pressing the "Toggle Controller Models" button. Similarly, the wizard can remove all extra visuals (i.e., user visual representations) from other user's environments by pressing the "Hide Extras" button.

Photon Controls The wizard is able to connect and disconnect from Photon, which is the networking backend for this system. This allows them to test interactions locally and to test issues with internet connectivity. Additionally, the wizard can check a ten message average ping using the "Latency Check" button. The system's networking implementation is covered in Appendix A.3.

Anchor Management When the "Remote Client Tools" button is pressed, an additional toolset is loaded, shown in Figure 3.18. This tool-set is used to change a single user's coordinate synchronization anchor rotation and position, to remove player trackers or visuals, and to change their PID. These controls are used by selecting the desired user from a drop-down and using the text entry fields or pressing the "Remove selected player trackers" button. Using the text entry fields changes the desired user's coordinate synchronization point or PID.

Annotation management In addition to being able to use the standard annotation interactions, the wizard is able to load, remove, delete, and save all annotations. These controls are shown in the second column of the menu in Figure 3.17.

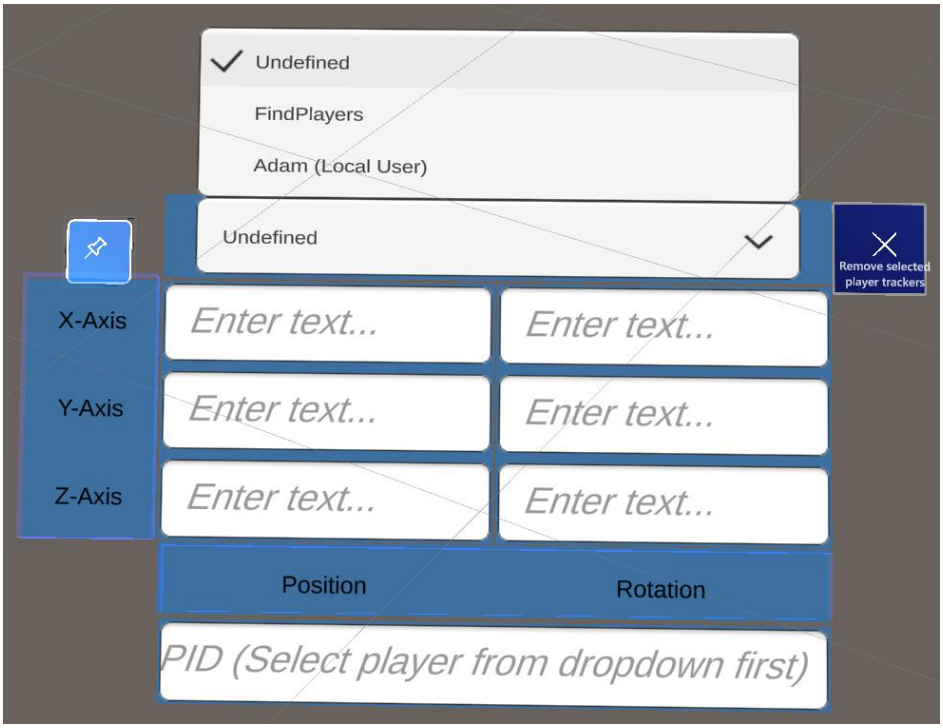


Figure 3.18: Controls for managing remote user's coordinate synchronization anchor, PID, and visual representations.

Logs A debug log and a photon log can be generated using the two log buttons shown in the fourth column of Figure 3.17. The debug log shows all messages that are sent to the console (left side of Figure 3.19). This log can show the wizard or researcher what errors are being printed while wearing an AR or VR HMD. The development console that would normally display these messages is not accessible in AR or VR HMDs. These logs provide a better level of debugging ability to researchers. The other log provided is a Photon log where key network information is displayed (right side of Figure 3.19). This includes the number of connected devices, current ping, names of other users, and internet connection status. Both logs are scroll-able text menus with a label, follow user button, and exit button (Figure 3.19).

3.7.2 Remote User Presence

When the human "wizard" acts as a recognition system for participant inputs, the wizard must quickly determine what the participant's intended command was and execute it. Navigating a 3D visualization can be complicated and watching a video stream of the remote user alone is often

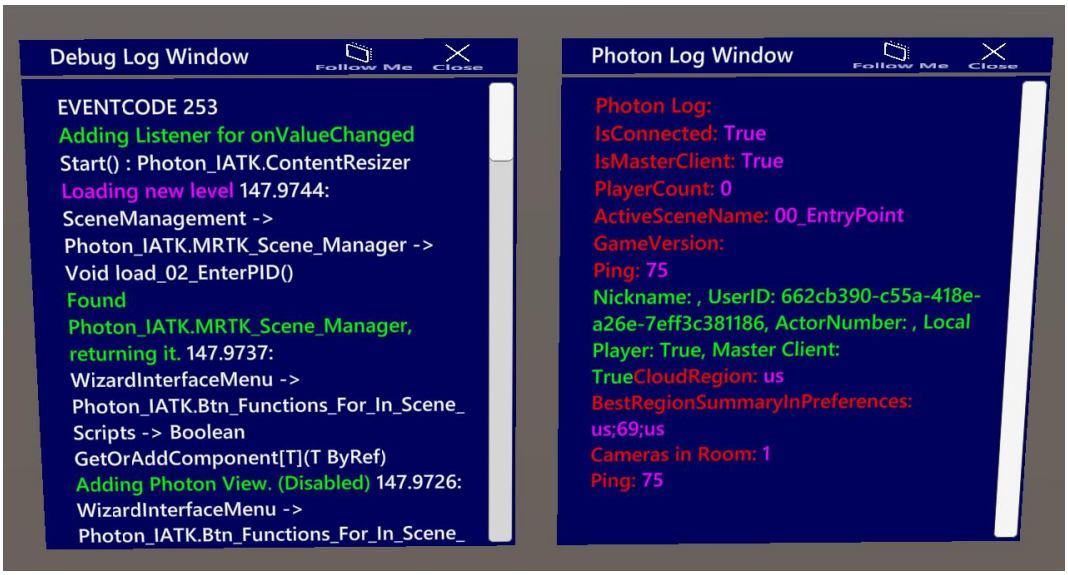


Figure 3.19: Log windows for the debug (left) and photon (right) logs.

not enough to make these predictions at speed. Several user presence visuals were added to the platform to improve the wizard's ability to interpret participant interactions (Figure 3.20).

Any connected platform will spawn a labeled "user representation" with location and rotation based on the user's head position (Figure 3.20). VR user's controllers are shown as models that match the positions of the actual VR controllers relative to the visualization. When someone logs into the platform in AR, the wizard is able to see red, green, and blue lines that represent the AR user's right-hand ray-cast, gaze direction, and left-hand ray-cast respectively (Figure 3.20). These lines will load an additional small green sphere and place it at any place a player ray-cast intersects with an object. This way a participant can say "move that" and the wizard will know what virtual object "that" is. In Figure 3.20, the remote HoloLens 2 user's right hand is pointing at a red sphere. The red line represents their ray-cast and the green sphere is the ray-casts intersection with the sphere object.

3.7.3 Wizard Interaction Feedback

To reduce interaction differences between a local user and a remote user (e.g., the wizard), the same visual interaction feedback is played for all connected parties. This means that when any user moves an object, all users can see both the visual feedback for translation and the object moving.

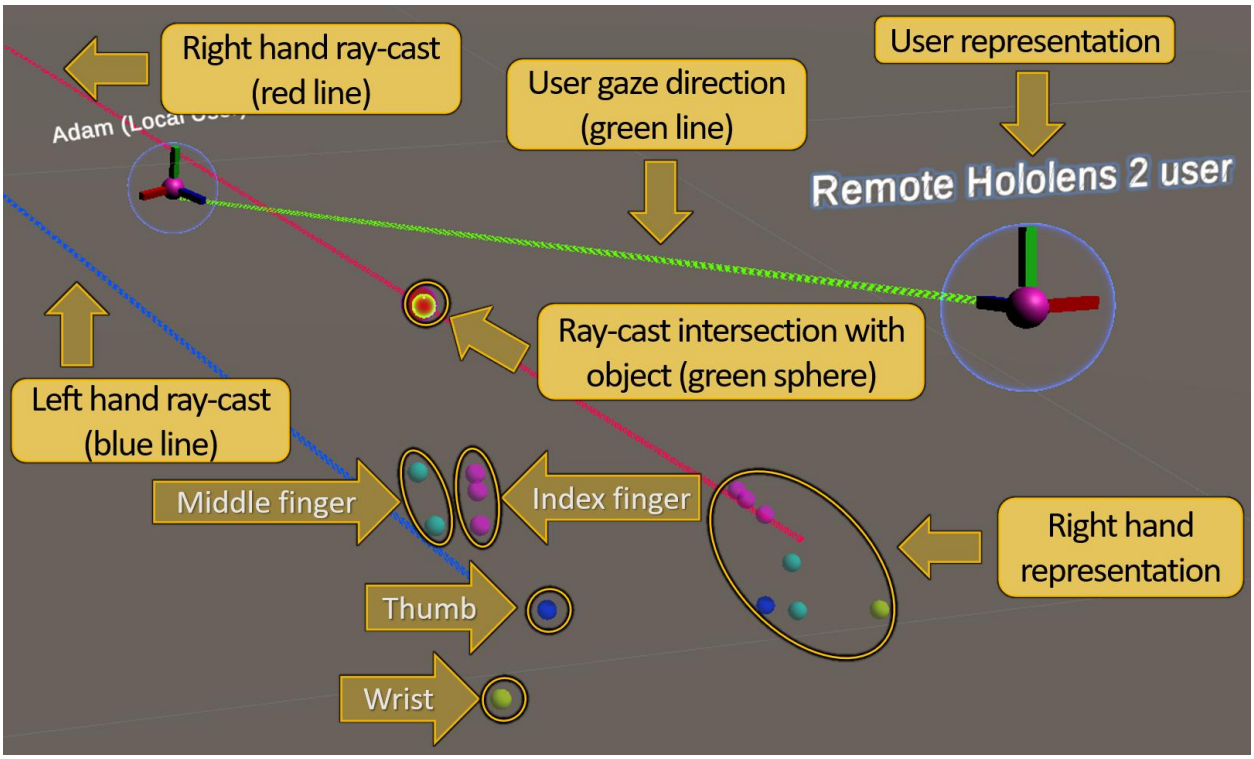


Figure 3.20: Labeled scene showing a remote HoloLens 2 user and a local player. The viewpoint is that of a third user.

3.8 System Design Conclusion

This system was designed to be usable by researchers working in different areas of IA (i.e., interaction design, annotation use, visualization markup). It is compatible across platforms allowing AR/VR comparisons. It can be used by multiple parties at the same time allowing for remote and co-located collaboration studies and elicitation studies and further enables cross-device collaboration. Another benefit of this system is its ability to save an intermediate state which can be loaded at a different time or by a different user. This allows researchers to investigate asynchronous collaboration and annotation re-visitation.

The next step in this system's development was to conduct a between-group usability and comparison study with both AR and VR users to provide insights both on how users interact in the environment and areas for improvement. This study identifies differences in how users navigate AR versus VR; differences that will assist developers in determining which interaction techniques are transferable between AR and VR systems.

Chapter 4

Cross Device System Evaluation

This chapter covers the AR-VR interaction comparison experiment conducted using the IA system discussed in Chapter 3. There is limited work examining interactions in IA environments [10, 11, Chapter 4] and even less exploring how people interact in an IA environment when using an AR-HMD. This study is one of the first works to implement a complex IA environment in both AR and VR. Leveraging that novelty, this is one of the only works that compares how users interact in IA environments while in AR vs VR. The results of this work provide insights into the paths forward for IA interaction research based on the compared user experiences between the two devices used (e.g., AR, VR).

4.1 Methods

This multi-platform IA system with annotation tools was developed and improved over the course of eight iterative design sessions spanning over 3 semesters of development. After those iterative sessions, four pilot studies were conducted to further refine the environment and research design. The pilots examined how people navigated the environment, what tasks they were capable of doing, and what tools would be used in the environment. After final changes were made to the system based on those pilots, 12 volunteers participated in an AR-VR IA interaction and tool use comparison study. These volunteers were randomly split into two groups of 6. Each group was assigned one device, either a VR-HMD or an AR-HMD. During this experiment, participants used provided tools for the analysis and annotation of a 3D scatter-plot. These tools include 5 types of annotations. A trash bin was provided to delete unwanted annotations and an "annotation station" was provided that participants could use to generate annotations and to change the visualization axis mappings.

4.1.1 Experiment Design

Sessions were conducted individually in a lab hosted by the university. Participants would arrive at the lab and were asked to sit at a 36-inch in diameter round table. On that table was a black mat with three image targets that were used by the system to align virtual content across devices. These image targets were each the size of letter paper. Each one had an image printed on it, these images were of wood chips, rocks, and asphalt.

Participants were given an informed consent form. After reading and signing the informed consent, participants completed a demographics survey, the short graph literacy scale, four questions about scatter-plots, and the VZ-2 paper folding test. After completing the pre-study forms, participants were told about the experiment at a high level and donned either the AR or VR HMD.

Once the environment was loaded onto their HMD participants completed an un-timed interactive training session during which, the researcher explained the environment. This training first covered loading, placing, adjusting, and locking the visualization, trash bin, and annotation station. Participants were asked to place these objects where they were comfortable with interacting with them acknowledging that object placement could be changed at any time.

After participant's work space arranged their workspace, they were told how to change the dimension mappings on the visualization (i.e., changing the x-axis). Once they felt comfortable changing mappings, the five provided tools were explained in order. These tools were the details on demand tool (DoD), the mean/median plane (centrality tool), a text box, and two highlight volumes. Participants were told that they could ask questions as they encountered them.

After the training session participants had phase one of the task explained to them. During phase one, they were instructed to navigate the visualization, examine the data, interact with the tools, and to generate questions about the data that could be asked to other users. Phase one lasted 15 minutes or until the participant asked for the next phase to be started.

Phase two was started at the end of phase one. Phase two consisted of a 15-minute session where the researcher would ask the participant questions about the dataset. These questions were always read by the experimenter and were repeated any number of times requested. Phase two

ended once participants answered all questions, once 15 minutes had elapsed, or when they requested to end the session. The main goal of this study was to observe how people interact with the environment, not to measure the accuracy of responses to these questions.

After the experiment, participants removed the HMD, completed a NASA TLX survey, and then answered questions in a semi-structured interview where the same set of questions were asked to each participant by the researcher. This was followed by any additional questions that arose out of the participant's responses or interactions while in the environment.

Participants were asked to think out loud during all phases of the experiment. Questions about interactions were always answered directly. Questions about how to get the answer for questions during phase two were given as hints indicating possible approaches to solving the question (i.e., what tools or settings could be explored). This choice was made to facilitate more interaction between the participant and the environment. As a result of this choice and the choice to ask varied difficulty questions the correctness of answers is not assessed.

The visualization used in this experiment was a scatter-plot graph showing a dataset containing the nutrition information of cereals. Some examples of this data include the dietary fiber or fat content that a given cereal has. This dataset was chosen for its simplicity and accessibility of subject.

4.1.2 Questions Asked During Phase 2

During Phase 2 of the experiment, participants were asked questions about the data represented by the visualization. The goal of these questions was to increase participant engagement with the system and to create a reason for participants to use the tools provided by the system. These questions were typically asked in the same order but could be asked in a different order if participants were struggling to answer them and could be repeated as necessary. If participants were struggling to answer a question, possible approaches to solving it were provided. This might look like suggesting that the participant changes the color/size mapping. If participants still struggled, another

clue would be given with more detail, an example being “try looking at the visualization with the color/size mapping to size fat”.

The first question asked to all participants was, “What manufacturer has the fewest cereals represented?” The visualization shows one manufacturer that has a single cereal represented. The rest of the manufactures had four or more cereals shown.

If participant’s answer to this question and others asked were nearly correct or was correct a more difficult question would be asked. An example of a nearly correct answer for that question would be naming the manufacturer with the second-fewest cereals produced. More difficult questions required comparison between two visualization states, identification of trends in the data, or required making inferences based on the data. The full list of structured questions asked is provided in Appendix A.2.

If participants were unable to answer the questions asked, they were asked “What is the highest sugar content contained in the scatter-plot” in place of a more difficult question. This question only required finding the highest value of a single axis of the visualization. If participants were unable to answer that question they were asked what the lowest value shown for one of their currently displayed axes.

These questions were not graded and the accuracy of their answers, they were only used as a mechanism for increasing participant engagement with the system. Asking the same set of questions during this task could have caused more disengagement in participants that were struggling to answer them. Due to this design, the accuracy of these questions is not assessed. Instead, the number of answered questions and the total time spent answering questions is compared between participants.

4.1.3 Participants

All participants were recruited using word of mouth and university email lists. In total, 21 participants volunteered to be a part of this research. These volunteers each participated in one of the following three types of study: iterative design, pilot studies, or the AR/VR comparison study.

Iterative Design

Five people, 2 females and 3 males participated in the iterative design sessions. Early volunteers participated in multiple design sessions allowing them to learn more about the environment and more thoroughly critique the system. Participants in later design sessions only completed a single session to more accurately capture the feelings of a new user to the system. The goal of the early iterative sessions was to polish the IA system to a usable state before running pilot studies. No demographic information outside of gender was collected from these volunteers.

Pilot Studies

Four volunteers participated in the pilot studies. These participants consisted of three males and one female with a mean age of 22.75 years and a standard deviation (SD) of 1.5 years. All of these participants were computer science majors that were either late in their undergraduate studies or early in their graduate studies.

AV/VR Comparison Study

Of the 12 participants in the AR/VR comparison study, two were volunteers, five were given $20 in gift cards, and five received in-class credit for their participation. The two volunteers were offered payment but turned it down for personal reasons. These participants were randomly split into two groups. All participants confirmed that they were comfortable interacting with 2D scatterplot charts and had normal or corrected to normal vision. The sample size of 12 participants was grounded in the common sample sizes of prior observational work [20, 119–121].

VR Group The VR group had an average age of 21.5 years with a SD of 3.99 years. Five participants had used an AR headset for 30 minutes or less prior to this experiment. All were right-handed. Two participants indicated that they played VR games. One participant played VR games for 3 hours a week and one played them for 1 hour a week. The VR group consisted of 5 females with 1 male.

AR Group The AR group had an average age of 25 years with a SD of 4.78 years. Three participants had used an AR headset for 30 minutes or less prior to this experiment. Four were right-handed. Two participants indicated that they play VR games for 1 hour a week. The AR group was composed of 5 males and 1 female.

Gender Imbalance There was a gender imbalance between the AR and VR groups. Ten males and two females were originally scheduled from this study and were semi-randomly split into an AR and VR group. This process was semi-random because gender was balanced between the groups, but within each reported gender participants were randomly assigned a group. Of these participants, seven either canceled their sessions or never showed up for their sessions. This caused a second round of word of mouth and email recruitment to be done. At this stage, the AR group was nearly complete having run one female and four male participants. One of the volunteers from the later recruitment was placed as the final member of the AR group and the rest were assigned to the VR group. This change in participants caused the resulting AR/VR groups to be either male or female-biased.

4.1.4 Apparatus

This experiment was conducted using two different platforms. For the VR group, participants used an HTC Vive Eye Pro. The Vive was connected to a Windows 10 computer with 32 GB of RAM, an Intel i9-9900k CPU (3.60 GHz), and an Nvidia 2080ti with 14 GB of memory. The AR sessions were conducted using a Microsoft Hololens 2. The system was developed using Unity version 2019.2.18f1, the MRTK version 2.5.1, Vuforia version 9.6.3, and the IATK [1]. This platform was developed on a Windows 10 computer with the same specifications as the computer used for the VR group.

4.1.5 Surveys Used

The Short Graph Literacy Scale was administered to all participants [122]. This scale consists of 4 questions that can be used to assess graph literacy and understanding. These questions

asked participants to interpret of a bar chart, a pie graph, a line graph, and pictorially represented quantities. These questions were taken from the full 13 question Graph Literacy Scale which was validated on 495 German citizens and 492 United States citizens [123]. The short version was determined to be a psychometrically valid method for assessing graph literacy based on an ANOVA comparison done using the same data as the original study [122].

Four additional questions specific to this experiment were added to the scale. Those questions asked participants to interpret a scatter-plot and provide the second-highest value, the correlation type, and the number of points above a set value. These questions were designed to mirror questions asked during the experiment. The short graph literacy scale with these questions is referred to as the SGLS+. Instructions for accessing the original SGLS and images of the three addition scatter-plot questions asked are provided in Appendix A.4.1.

All participants completed the VZ-2 paper folding test. This test assesses visual and spatial reasoning skills by having participants determine where a hole(s) would appear in an unfolded piece of paper when a pencil is pushed through a folded piece of paper. This test consists of two three-minute sub-tests. The paper folding test, or hole-punch test, was originally introduced in the "Kit of Factor-Referenced Cognitive Tests" in 1976 [2] and has since been used by many researchers [124], including those in researching IA [120, 125, 126]. The version of the paper folding test used here is provided in Appendix A.4.2.

At the end of a session, participants completed a NASA Task Load Index (TLX) to measure their perceived workload when interacting with the IA system [35]. The NASA-TLX is a survey used to rank perceived workload across six subcategories; mental demand, physical demand, temporal demand, performance, effort, and frustration, which are then combined to create an overall score [35].

4.2 Data Collection

Video data was collected using a web camera that was set up in front of the participant and by using the HMD's onboard cameras. This video captured the participant's speech and an exo-

centric view of their body. Video from inside of the running system was captured from the point of view of the participant. This video included the virtual environment and in the case of AR the real environment. Because capturing video in AR is prone to error or disconnection, an additional video of the participants' interactions in the environment was recorded from the perspective of another user. That video was only used in the times where the AR video was not collected due to an error. All participant commentary was collected using a microphone that was recording alongside the video. Open Broadcaster Software [9] was used during the sessions to merge the different video and audio streams into a single .mkv output. This merged video showed a set of four labeled videos that were arranged in a two by two grid. These videos were synchronized and represented the environment video, the webcam video, the device video, and in the case of AR, the backup video. This merged video allowed viewing of all videos streams in a single media player, preserving their time alignment.

The IA system automatically collected log data for all events in the system including any object manipulations, participant movements, scatter-plot changes, and annotation tool interactions performed. This data was recorded with a row for every event that occurred during a session. These rows were timestamped as <YearMonthDate-HourMinuteSecondMicrosecond>. The rows always included what the event was (i.e., translation), who initiated the event (i.e., the participant), the object affected (i.e., the visualization), and the current state of that object including its position, rotation, and scale. These logs were output as CSV files that were actively written while the session was running.

The NASA TLX, SGLS+, and demographics surveys were collected on a computer and saved as .csv files. The paper folding test was administered on paper and later transcribed to a .csv file.

4.3 Data Preparation

Video data was watched by the experimenter while taking notes of comments made by participants, how the participant was interacting, what the participants were struggling with or excelling

[9] https://obsproject.com/

at, and the timestamps of all major events (i.e., session start, phase one start). The merged videos were watched using DaVinci resolve 17 [10] a free to use video editing software that allows play speed adjustment, rewinding, fast-forwarding, and the partitioning of videos. Notes were recorded in a text editor.

Log data was combined across participants and cleaned using R in combination with R-Studio [11]. The cleaning process involved removing any actions not made by the participant. Most commonly these actions showed up as rows that were logged when other parties joined the session. In the case of AR, when the VR-HMD loaded the environment to record the backup video, a number of system events were triggered. These events were removed from the data by removing any rows that were not initiated by the AR-HMD using the event "caller" number. In VR, the participant did not directly load the environment because it was an application on the experimenter's computer. In these cases, the experimenter would start the session just prior to the participant arriving. This caused log data to be generated using the same caller number as the participant. These rows were removed by removing any rows that had a time difference of more than five minutes between them. All motions in this system including those of controllers and headsets generate a row in the data. This means that there were no time gaps larger than a few seconds between rows allowing removal of these initial rows and any tailing rows. Tailing rows occurred after the participant takes off the headset but before the system is shut down.

4.4 Data Analysis

After being cleaned, the log data was used to generate files for each type of manipulation (i.e., translation). Within a file, the data was averaged over participants and objects such that the position log included a row for each participant and each object type. Object types were annotation, visualization, and other. Other covers the annotation station and trash bin. In total this generated 12 (participants) X 3 (objects) rows of data. To ensure consistency between sessions with different

[10] https://www.blackmagicdesign.com/products/davinciresolve/

[11] https://www.rstudio.com/

time durations, interaction data was divided by the total time of the session. This data is reported here and includes information like the average degrees of visualization rotation per minute for a given participant's session. Another file was generated with the aggregated data of the visualization state changes. This data was a binary true or false for whether a participant had used that graph state or not. The NASA TLX, SGLS+, and paper folding test data were interpreted using the average per system used (i.e., AR, VR) and per participant. Each of these files consisted of a row per participant and a column per feature. For the visualization state log, this produced a 12 row by 14 column dataset. There was one column for each dimension mapping available and one row for the participant ID.

Results are reported using box and whisker plots where possible. The mean and SD values for data are provided for numeric data. This was done to maximize the transparency of the reported data. This environment is unfamiliar to participants and there is limited prior work in this area. This data includes wide variations between users within an individual device and between the two devices. This data was not interpreted using more in-depth statistical analysis due to the limited sample size and variation between individual's interactions.

Participants in the AR group fall into two categories. There were three participants that completed the full experimental sessions, interacted with most or all of the tools available after the training session, were able to answer questions about the data set during the second phase of the task, and viewed most or all of the possible states that the scatter-plot visualization could be set to. The three other participants ended at least one phase of the experiment early, were unable to answer most questions asked, and struggled to navigate the environment causing most visualization states to not be used. These two groups are referred to as the top performers and the bottom performers. In the VR group, there was not as clear of a division of abilities. As such the top and bottom performer labels are only used in the AR group. The three bottom performers were not labeled as outliers due to the limited information available on user performance and interaction in AR IA environments.

4.5 Results

4.5.1 Iterative Design Sessions

The results of the iterative design sessions were worked into the environment prior to running the pilot studies. The first few iterative design sessions led to the implementation of buttons over drop downs for graph navigation, visual feedback for manipulations, audio feedback for manipulations and button presses, and a more streamlined experiment set up where participants need only press three buttons to set up a session in AR and 2 buttons to set up one in VR. These improvements emerged from participant commentary on what they were struggling with when interacting with the system. More details on the changes made to this system out of the iterative design sessions is given in Chapter 3.

4.5.2 Pilot Studies

The pilot studies were beneficial in determining the direction of the experiment and the refinement of several of the tools used in the environment. The suggested improvements from the pilot studies were incorporated into the system prior to running the first participant of the full AR VR comparison study.

Changes to the Instructions The first 3 pilots used video instructions to train participant on how to use the environment. These instructions were 7 minutes long and covered all aspects of the environment that participants would encounter. These participants were unable to remember most of the system functionalities by the time they loaded the environment, causing them to ask for guidance on how to navigate the system. These participants reported that there was too much information for them to retain from the video instructions.

The solution to these complaints was to provide an interactive training session in the environment where the researcher went over each system functionality with the participant while the participant performed the actions being explained. This was tested during the last pilot and found to improve the participant's understanding of the environment during the experiment seen by a re-

duced count of interaction questions asked and an increased use of the tools available in the system. Even so, that participant still asked questions about interactions in this environment indicating that the interactive training improved the participants ability to navigate the system, but also that the system was complex enough that users may not be able to perform all actions in the system after a single training session. Interactive training was used in the full AR VR comparison study and participants were told they could ask questions whenever they had them.

System Control Improvements While interacting with the system, pilot participants commented that returning the visualization to a level state using the provided manipulation handles was difficult. They did not want to adjust each axis individually. To resolve that a button was added that would automatically level the visualization in reference to the real world ground.

A second common struggle observed during the pilot studies occurred when participants attempted to press the buttons on the provided annotation station. When participants used the selection interaction on their device to execute a button press, it was likely that their hand would move a little. If participants were not mindful of this movement, the system would interpret it as a move command cancelling the button press command. The issue of accidental movement was also observed when participants attempted to interact with an object above the visualization (i.e., a highlight cube) but missed the object and grabbed the visualization instead. To resolve both of these issues a lock button was added to the system that would disable translation interactions on the annotation station, visualization, and trash bin. This drastically improved participants ability to press buttons and reduced the number of accidental chart movements that occurred.

These buttons were added to the participants' button menu after the second pilot, allowing them to be tested in the latter two pilot studies. The participant button menu also housed the buttons used to set up the experiment and center the visualization.

Wizard of Oz Design The pilot studies used a WoZ design where the participants were able to use the same ray-cast controls as in the full study and were told that the system had speech and gesture interactions as possible inputs. It was important to allow participants to use basic inter-

actions in this environment to increase their immersion and their belief that they were interacting with a fully functional system. Some interactions in 3D space are difficult to emulate as a wizard. One example of this is seen when trying to move an object along a path specified by the user. This path specification will be different from the path that the wizard actually moves the object because of the difficulties found in moving objects in 3D space and in interpreting participant intent in a complex environment. Allowing participants to control object movements resolved that concern. Other commands, like changing and axis or loading a tool, are easier for the wizard to execute. The combination of live interactions and wizard interactions was intended to maximize participant engagement and reduce some of the participants' difficulties with graph and object management.

Participants were told that they could use additional inputs to interact with the system (i.e., gestures, speech) during the video instructions and during the session. These controls were brought up a second time after participants donned their HMD. None of the pilot participants used commands that the wizard would need to interpret. In actuality, telling participants about how they could use these non-standard inputs confused them. With no pilot participants using the wizard for input recognition and with the difficulties participants already had understanding how to use the basic interactions in the system, the WoZ design was removed from the final experiment.

Mid-Air Pen Changes A line drawing tool in the form of a VR-Pen was originally included in this study. This tool allowed participants to draw lines on physical surfaces or in mid-air. Two of the pilot participants used the mid-air pen. The other 2 participants only used the mid-air pen at the conclusion of the session after being prompted to by the researcher.

The first participant used the pen frequently at first, but stated line size should be smaller to more closely resemble a normal pen. Out of this the pen size was reduced. The second major issue encountered during the first pilot session was the accidental drawing of small marks which would then occlude objects behind them, thus preventing those objects from being selected. The issue of occlusion was resolved by removing the larger ray-cast collision bounds on the drawn line leaving only a small ray-cast collision bound that matches the actual placement of the line. When the line was intersected with the ray-cast the larger bounding volume with manipulation controls would

appear. The next improvement made was to remove line segments that were 12.7 millimeters or smaller automatically to reduce the number of accidental marks on the visualization.

The second participant to use the pen attempted to write a note but was not able to use small enough hand writing due to a combination of the difficulty of writing in mid-air and the system not accepting pen strokes smaller than 12.7 millimeters. That participant did not use the pen again. In response to this occurrence, the minimum line size was reduced from 2.3 millimeters to 0.0635 millimeters.

The final two participants noted that the pen worked after being prompted to use it, but that they would not use it in this environment because making marks in mid-air did make sense to them. Those participants believed that marking paper made more sense than drawing in a virtual environment.

Ultimately the mid-air pen was removed from the final study design. This decision was based on the limited use of the pen during the pilot studies and the difficulties that participants had learning the other annotation tools in the system. Removing the pen tool reduced the complexity of interacting in the system and allowed participants to have both hands free where with the pen they would need to have one hand on the pen or place it somewhere on the desk in front of them.

Annotation Tool Improvements Over the pilot studies, participants voiced interest in a tool that could count the data points in a given area of the scatter-plot. This tool would help them overcome some of the difficulties faced when counting objects in 3D space where object depth can be hard to determine. This feature was added to the highlight volumes which already had access to that information due to their need to highlight the points encapsulated within them. With this addition the highlight volumes displayed a text label above them that showed the count of points encapsulated inside of them.

Originally, the details on demand tool generated tick marks that attached to the axis and matched it's position in the data. This provided a way to see the position and depth of the DoD tool when using it inside of the visualization. It also allowed participants to mark points on the axes by placing the DoD tool near them. In the pilot studies, one participant found that seeing the

information for the actual data point in the scatter-plot nearest the DoD tool was more beneficial than seeing the data for the location of the DoD sphere on the tick marks. To facilitate that the tick marks generated by the DoD tool were changed to match the location of the real value data point nearest the DoD tool.

The last change to the annotation tools was done to the centrality planes. These planes have a button panel on their lower left side. This panel had the controls for changing the axis and centrality metric that the plane was set to. If the plane was set to the y-axis or z-axis the panel would face either up or left making interactions with it from the front of the visualization difficult. In response to that, this panel was set to face the users head position so that they could always press and see the panel. The plane would still orient its self along the axis that it was set to such that when set to the y-axis the plane face was parallel to the bottom of the visualization.

4.5.3 Paper Folding Test

The VR scored slightly higher than AR group for spatial reasoning ability with lower variation in their scores. The mean score for paper folding test in the VR group was 74.14% with a SD of 12.39% where the AR group had 72.50% (SD 20.36%). The AR group score variation was caused by two participants receiving low scores, one receiving a 50% and the other a 40%. The lowest two scores in the VR group were a 55% and a 65%.

The participant that scored the 40% in the AR group was one of the 3 top performers in that group while the participant that scored the 50% was one of the 3 bottom performers. These differences in performance indicate that spatial reasoning may not be a strong indicator of participant ability to interact in this environment.

4.5.4 Short Graph Literacy Scale Plus

On average the AR group scored higher than the VR group at both the SGLS and the 3 additional scatter-plot questions. The scores for the SGLS were 92.86% (SD 11.29%) for the AR group and 75% (SD 14.43%) for the VR group. The scores for the 3 additional scatter-plot questions were very close between the two groups. Four out of six people in the AR group scored 100% and

two out of six scored 66.67% where half of the VR group scored 100% while the other half got a 66.67%. This brought the average score for the AR group 92.86% (SD 15.06%) and the average score for the VR group 83.33% (SD 16.67%).

In the AR group there were top and bottom performers that scored 100% on both sections of the SGLS+. The AR group outperformed the VR group at the SGLS while also having more participants struggle to finish tasks in the environment. This indicates that the SGLS and the additional 3 scatter-plot questions may not provide a clear signal of a participants ability to navigate stereoscopic IA environments.

4.5.5 Experiment

Time Spent in Environment

The AR group spent more time in the training portion of the experiment and less time in the rest of the experiment (Figure 4.1). This was caused by participant request to end either phase one and/or phase two early. Typically these participants seemed less comfortable in the environment, and were less likely or less able to answer questions asked about the visualization. This was most seen in two participants of the AR group who both indicated that they wanted to end both phase 1 and phase 2 early, most likely due to their difficulties with interacting in the system. These participants struggled to correctly select or interact with objects in the environment, causing their inclusion in the bottom performer group.

Other participants in the AR group ended the sessions early but only by a few minutes each. Two of these participants stated that they could not generate more questions to ask about the data, indicating they wanted to move on to phase 2. Participants in the AR group also struggled to answer questions during phase 2. After failing to answer 4 questions in a row two participants became very disengaged with the task, stopping their interactions with the visualization. The VR group had less deviation in their times and did not have anyone request to end a session early.

Participants in the VR group finished the training phase in 8:14 minutes on average compared where the AR group took 12:53 minutes on average. One participant in the VR group skipped

four of the questions asked which resulted in a phase two duration of 12:06 minutes. Another VR participant answered all questions in under 15 minutes ending phase two in 13:08 minutes. Interestingly two of the VR group participants chose to stay in the environment longer than 15 minutes during phase 2. One of these participants was interested in the environment and the other was determined to answer the last question asked. These two participants stayed in phase 2 for 23:18 and 17:56 minutes respectively.

The bottom performers in the AR group exhibited similar tendency's as each other. Bottom performers took nearly twice as long as the top performers to complete the training session (16:43 minutes compared to 9:03). They also spent less time in the first and second phase, often stopping all interactions with the visualization and system towards the end of each phase. The top performers in the AR group and all of the VR group continued interacting with the environment until and at time past when the phases ended.

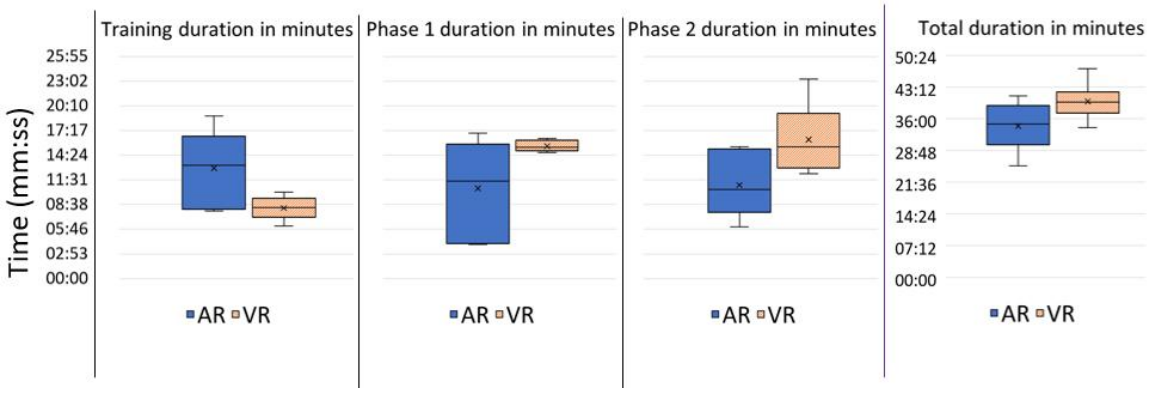


Figure 4.1: Times that participants spent in different portions of the experiment by device condition.

Phase 1

During phase 1 participants were asked to generate questions about the dataset and to explore the dataset. The only constraint given was that these questions needed to be ones that could be answered using the tools provided by the system. Only two participants in either group generated questions. In the AR group these participants generated 9 and 4 questions where in the VR group they asked 1 and 4 questions.

The AR participants were much more likely to stop interacting with the visualization during phase 1, seen with the three bottom performers. These three AR participants became frustrated with the system and chose to not ask questions about the data. These participants also interacted with the visualization and tools less than was seen in rest of the AR group and all of the VR group.

Phase 2

During phase two participants were asked questions by the researcher. These questions were repeated any number of times and could be skipped by participants. Easier questions were provided to participants that were struggling to use this environment in order to encourage more interaction with the system. The maximum number of questions asked was 11 with a minimum of 3.

In AR participants answered an average of 5.83 questions (SD 2.67) where in VR participants answered an average of 9.5 questions (SD 1.26). Participants in the VR group were able to answer more questions than the AR group even when only the top performers from the AR group are compared in isolation. These top performers answered an average of 8.33 questions (SD 1.25).

Three AR participants were unable to answer multiple questions causing the session to end early, these participants answered 3, 3, and 4 questions. One participant in AR skipped 3 questions resulting in 8 answered questions and ending phase 2 early. In VR all participants continued to answer questions until the end of the session or until 11 questions were asked. One participant in the VR group chose to stay in the environment longer to answer the final question asked resulting in them staying in the environment an extra 8:18 minutes.

Participants that were more active in exploring the data during phase 1 were more able to answer questions during phase 2. One participant in the AR group was able to answer one question based solely on their memory of the scatter-plot. In both groups participants had difficulties finding the correct approach to answering questions that required comparison across two different visualization states. There were not notable differences in the approaches taken for answering questions between the two groups.

4.5.6 Visualization Size

The visualization always began as a .42 X .42 X .42 meter cube. Participants were aloud to resize the visualization at any time. Most participants would resize it a few times in the beginning of a session then leave the visualizations scale alone. The average sizes of the visualization over the duration of the experiment is shown in Figure 4.2.

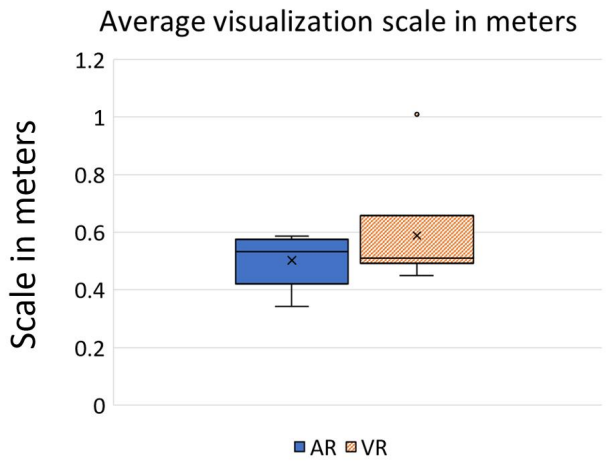


Figure 4.2: Average size of the visualization in meters by device used.

The VR group used larger scatter-plots than the AR group with an average visualization size of .588 meters (SD .191) compared to .502 meters (SD .085). The largest visualization used was 1.01 meters, used by a participant in the VR group. With this size visualization, annotations needed to be moved further than with a smaller sized visualization. This scale also increased the level of precision needed when interacting with annotations and manipulation handles. The participant also placed their visualization further away than other participants. When asked about the visualization size after the session the participant did not believe that the visualization was that large, possibly due to the combination of it being both larger and further away.

People across the VR condition tended to sit further back from the visualization and keep their hands closer to their body. The AR condition could see the desk in front of them and all participants in the AR group placed the visualization on that desk. Some of the differences in scale might be caused by the lower field of view provided on AR-HMDs compared to VR-HMDs.

When using the AR-HMD participants would need to use a smaller visualization to see the entire visualization at once. No participants commented on the field of view being a contributing factor to their interactions in the environment.

There was an interaction technique for scaling that was not covered in training. This technique involved selecting a single object with both hand's or controller's ray-casts then either pulling them apart (enlarge) or pushing them together (shrink). One participant in AR and two participants in VR found this interaction naturally. Another participant in VR asked if uniform scaling was possible and was told about this interaction. When the participants that spontaneously discovered this interaction were asked how they came across it the most common response was that the interaction seemed intuitive (105, 111, 109). One participant followed up with it was likely due to pop culture and the Marvel *Iron Man* movies which featured a interactive mid-air gesture system.

4.5.7 Visualization and Annotation Movement

On average the VR group moved both the visualization and annotations more per minute of the experiment than the AR group (Figure 4.3). The VR group moved the visualization an average of .859 meters per minute (SD .716) where the AR group only moved the visualization .334 meters per minute (SD .344). Annotations were also moved more by the VR group more than the AR group, with an average movement of 4.05 meters per minute (SD 3.832) and 1.198 meters per minute (SD .55) respectively. If the participant with the largest visualization is removed from the VR group, the VR group moved visualizations .682 meters per minute (SD .654) and annotations 2.37 meters per minute (SD .77) which was still more movement than the AR group. With the VR group sitting further away from their visualization and and using larger visualizations it makes sense that they would need to move objects further per minute. This is seen in the difference between moving an annotation across a .5 meter visualization and a 1 meter visualization.

Three participants in the VR group moved the annotation station from where it was loaded on their left to their right. These were right handed participants. No participants in AR did this. In VR participants were more likely to place the annotation station above their shoulders where

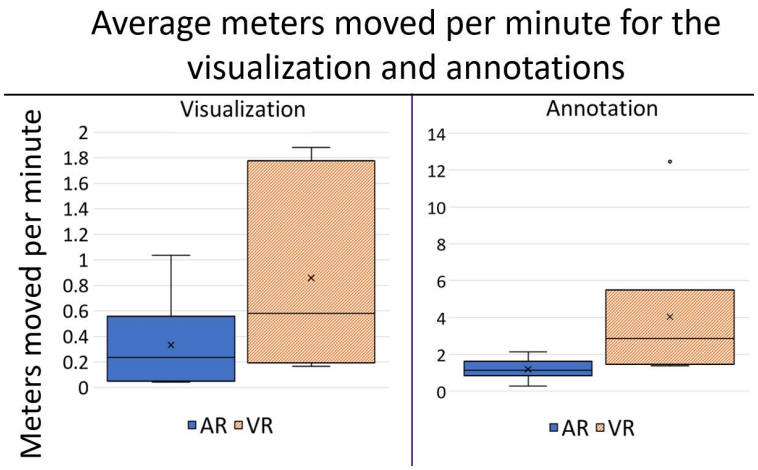


Figure 4.3: Left: Average visualization movement per minute spent in the environment, Right: Average annotation movement per minute spent in the environment. Y-axis units are meters.

in AR the annotation station was placed near the surface of the physical table in-front of them. Further differentiating the placement of objects in VR compared to AR, likely due to the real-world affordances seen by the AR group.

4.5.8 Visualization Rotation

In this environment an x-axis rotation is pitch, a y-axis rotation is yaw, and a z-axis rotation is roll. Figure 4.4 shows the differences between rotations in degrees per minute (Deg/Min) for the AR group (left) compared to the VR group (right). Rotations about the x-axis were the least performed rotation for each group with a mean of 12.66 Deg/Min (SD 3.64) for the AR group and 11.24 Deg/Min (SD 7.10) for the VR group. Rotations about the z-axis were the next least used rotation at 17.06 Deg/Min (SD 10.06) for the AR group and 13.40 Deg/Min (SD 8.52) for the VR group. The most performed rotation was yaw or rotating the visualization about the y-axis. In AR participants performed more yaw rotations with an average of 32.98 Deg/Min (SD 27.90) compared to 23.19 Deg/Min (SD 10.96) for the VR group.

Participants in AR were more likely to use the rotation handles than participants in VR were. In place of rotating the visualization, VR participants noted that it was easier to move their head. One participant added that moving their head enabled changing their view of the visualization on multiple axis at once where rotations using handles performed single axis rotation. This trend was

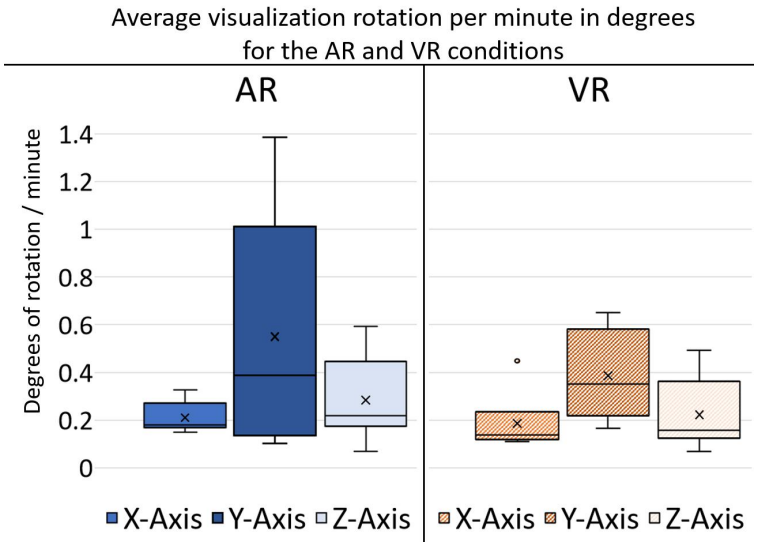


Figure 4.4: Average rotations in degrees per minute performed for each axis by system condition

also observed in the video recordings of the sessions which show that people in AR were more likely to rotate the visualization to see alternative views where in VR participants were more likely to move their body to see different views. Most often this head movement included leaning in and turning to see the side of the visualization; however, 3 VR participants stood up at one or more points to view it from above.

One participant in the AR group chose to leave their visualization unlocked thus allowing ray-cast translation interactions. When moving a object rotating the controller or hand causes a roll rotation which that participant used in place of the manipulation handles. This participant remarked that it was easier to interact with the visualization this way because they no longer had to use the handles.

4.5.9 Visualization States

The visualization had 3 axes with 2 mapping options each and 7 color mapping options including no color. VR group participants were more likely to see all of the visualization states than those in the AR group (Figure 4.5). In the VR group there were 3 participants who did not see all states with 2 of those participants missing one state, and 1 participant missing 2 states. All of the VR group's missed states were in the color/size dimension. In contrast to that, every participant in

the AR group missed at least one visualization state. Two of those participants were in the bottom performer group and did not view each axis mapping. Most of the AR participants missed 3 or more color/size mappings with limited differences in color/size mapping use between the top and bottom performer groups.

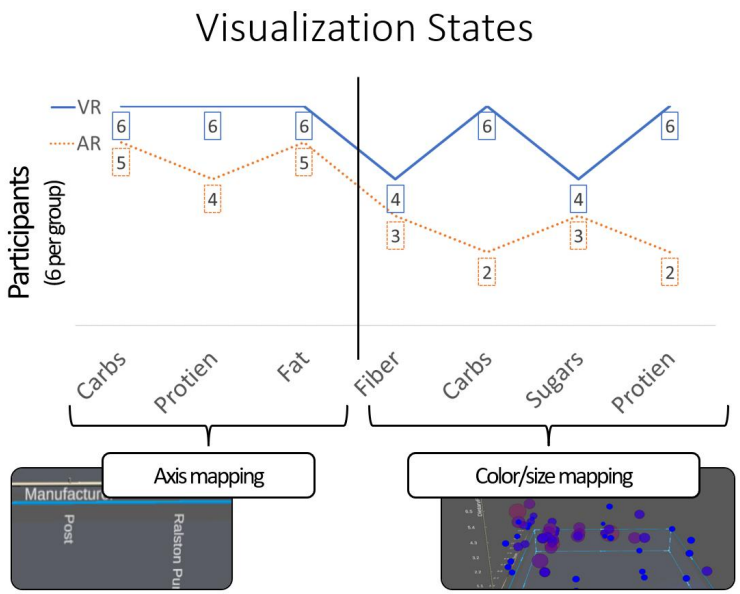


Figure 4.5: The number of visualization mappings seen by participants in the AR and VR groups. Mappings that were used by all participants in each group are excluded from this figure.

4.5.10 Participant Interactions

Both groups struggled to complete interactions using the ray-cast selection technique. In AR these struggles were seen twice as people had to move the ray-cast by moving their hand and select by pinching their fingers. Some users had a difficult time executing the pinch gesture in a way that was recognizable by the AR-HMD. In VR participants moved the ray-cast with a controller and selected by pressing a thumb button on the controller. In both groups trigger selection and natural jitters in arm movement caused inaccuracies in selection.

VR participants tended to interact with further away visualizations and held their hands closer to their body. The AR group interacted more directly with a closer visualization. With the visual-

ization being within reach, AR participants often held their hands near the visualization and thus further from their bodies.

A few participants in the AR group used their finger to press the buttons on the annotation station, allowing for a closer placement of the annotation station. However, most AR participants used the ray-cast instead of their finger when interacting with buttons. VR participants placed the annotation station further away and higher than the AR participants.

4.5.11 NASA TLX

The NASA TLX results are shown for each score category as box and whisker plots comparing across AR and VR in Figure 4.6. Also shown in that figure is line chart comparison between the means of the scores for each category (bottom right of Figure 4.6). In general there were limited differences between AR and VR conditions seen in the plots for frustration (38.33 AR vs 36.67 VR) and overall workload (55.42 AR vs 52.08 VR). These low scores for frustration and overall workload are interesting when considering the differences in interaction techniques between the two devices. These scores imply that the selection technique (i.e., button vs pinch) and the ray-cast movement type (i.e., controller vs hand) did not contribute to widely varied frustration scores.

The physical demand was more varied and slightly higher for the AR group than the VR group (mean of 46.67 AR vs 30.83 VR). This could be excepted as VR controllers can be used with less movement that AR mid-air gestures. AR participants perceived that they were using more mental and total effort than the VR group reported. This difference might be contributed to by the difference in engagement between the two groups. The VR group interacted with the environment more fully and longer than the AR group.

Participants in the AR group reported feeling a higher sense of performance than the VR group. This is contrasted by their actual performance, which was lesser in most categories than the VR group. Some of these performance differences can be seen in the time spent in the environment. The VR group felt less temporal demand which may have allowed for greater time spent in the

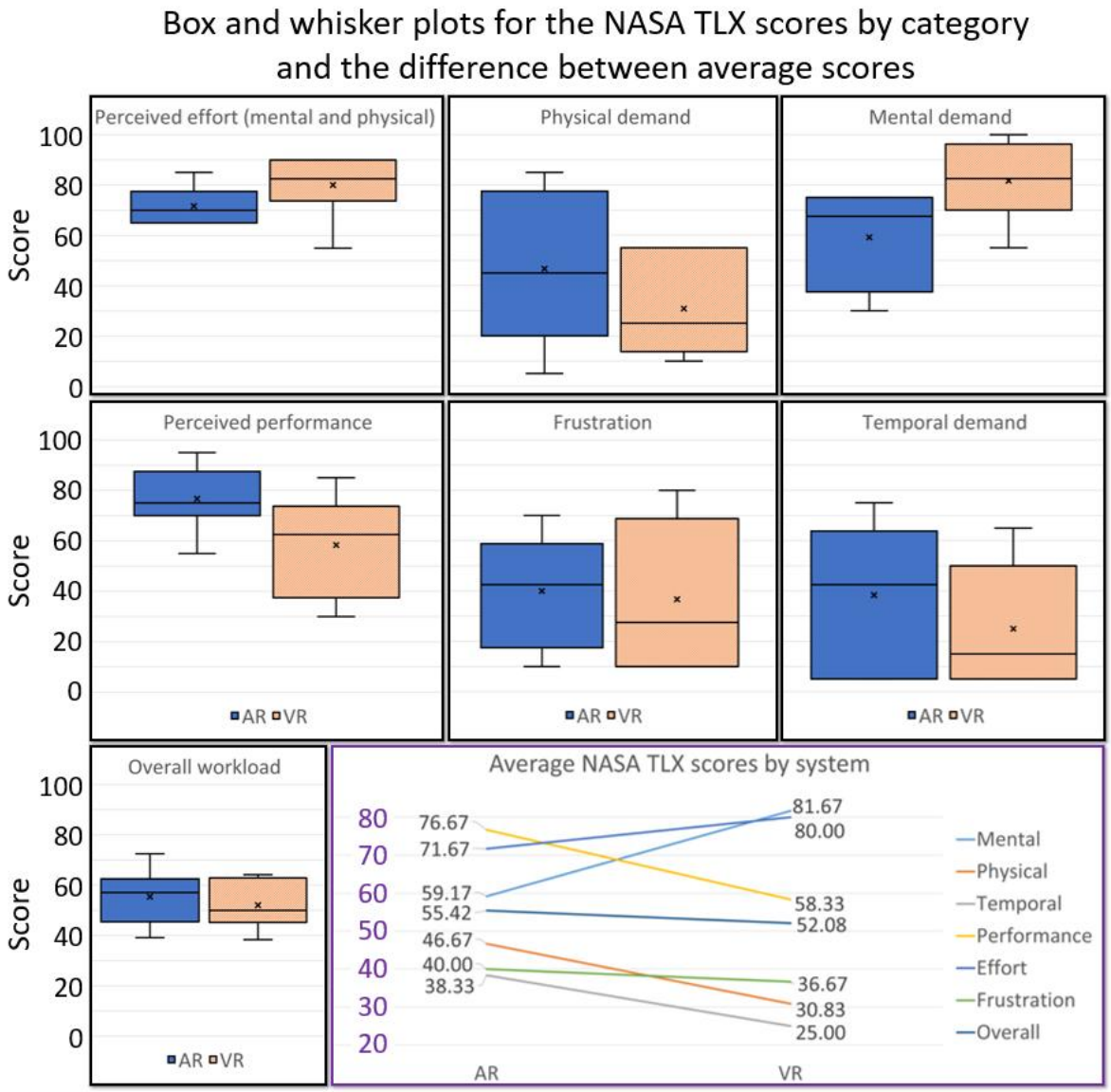


Figure 4.6: NASA TLX scores compared between AR and VR conditions. Box and whisker plots are provided for each score category and a line chart is used to show the difference between the average scores by score category and condition.

environment. Our speculation is that the AR group could see the real-world, including the experimenter, which may have caused them to feel more pressure when interacting in the environment.

4.5.12 Participant Interviews

All participants said that they enjoyed the environment, although many admitted that it was unfamiliar to them and thus difficult to use. This feeling was more prevalent in the AR group

where the 3 participants that struggled with interactions in the environment also had difficulties with answering or generating questions during the task. Participant 104 said that "[This system was] very different. I have been using 2D all my life so I am very used to it. This was unfamiliar. It was very fun that it can graph and plot a lot more things than I normally can and I can look at it from all angles which gave more information than a 2d plot [would]. If I was more used to it I could get more out of it... Once I have familiarity [with this system] I could draw a lot more conclusions from it [than a 2D system]". This comment's sentiment was echoed across several participants in both groups.

Several participants in both conditions noted that the system was nice in that they could use as much space as they want when setting up the visualization and placing annotations. This did come at the cost of some things being able to be moved very far from the user. In response to that two participants in the VR group wanted to have a more bounded system.

The VR group noted issues with depth estimation. These participants had a difficult time determining where on the z-axis points were. These depth issues can be seen in the participant who scaled the visualization to the largest size encountered in the study and placed it furthest from them-self, yet were unaware that it was that large or far away. Another VR group participant used a highlight annotation to check their ability to estimate depth in the environment by counting the points in an area then using the highlight on the same area to check their count. The AR group could use real world visual cues to determine the relative positions of things in the environment, helping mitigate depth perception issues.

The AR group had more concerns with the color mappings, where lighter color points were more difficult for them to see. One participant in the AR group attempted to construct visual boundaries by scaling the centrality planes to a large size and placing them as a wall behind the visualization. This helped them more accurately perceive the colors of the points in the visualization. Visual clutter was also noted as an issue in the AR condition where one participant said that they felt the need to clear out the graph (i.e., remove annotations from) before moving on to the next question.

Features Requested

The most commonly desired feature was a tool that could slice the data (e.g., reduce the axis values displayed) which was requested by four participants. A few participants wanted labels displayed for the data points and one participant went a step further saying they wanted to be able to set which information was displayed similar to the size/color mapping.

Interestingly one participant in the AR group and one in the VR group mentioned wanting a line tool to circle points similar to the mid-air pen that was removed from the experiment based on the pilot studies. Others requested features including being able to scale the visualization the to size of a room (2 in the AR group), trend lines (3), speech controls (1 in the VR group), a button to remove all annotations (1 in the AR group), and a way to switch between 2D and 3D visualizations (2).

Two additional tool requests came from the AR group which were 1 participant requesting a knob or button system for controlling rotations and 2 participants requesting a color picker that would let them change the color mappings.

Thoughts on Annotation Tools

Details on Demand The DoD tool was well received by participants. Issues with it were that if was possible to move it too far away from the user. One participant suggested that the tool should be changed to show a single detail panel and to have the details show for the nearest point if it was close to one. Another two participants did not like the axis bars saying that they showed to many details at once. A suggestion for improving the movement of the annotation was to allow locking it to a single axis of movement.

Centrality Planes Participants thought that this tool was very useful, all liking that it would automatically provide the mean or median value for a given axis. The most common complaint on this tool was that it is difficult to select the buttons on it. This was caused by movement of the ray-cast after the selection action is triggered which caused the objects to move overwriting the button press command.

Text Text was the least used tool. Five participants said that they had no need to use it. Other participants said that using it was too difficult due to needing to correct mistyped words and needing to use a virtual keyboard. That participant wanted it to delete full words at a time instead of a letter at a time when using the backspace button on the virtual keyboard. One participant commented that the text box would be beneficial in situations where they had to make a presentation on the data.

Highlight Volumes The difficulties with using the manipulation handles, and in particular, using them for scaling, was the most commonly mentioned issue with the highlight volumes. The scaling difficulties made it hard for participants to highlight regions of desired points on the scatter-plot. Suggestions for improvements included wanting a highlight plane, similar to the centrality planes and adding a different mechanism for scaling. One participant wanted to be able to uniformly scale the highlights.

Participant Thoughts on 2D VS 3D Visualization Systems

Participants felt that interacting in 3D was beneficial when they needed to compare across more than 2 dimensions of data. The bottom performers in the AR group would prefer to use a 2D system for a general data exploration task, feeling it would be easier to use because they already had familiarity with it. Another bottom performer noted that they would want to use a 2D system but acknowledged that a 3D system could provide more information at once. They mentioned having a hard time visualizing in 3D which made working in 2D easier for them.

The other nine participants said that they would prefer to use a 3D system over a 2D system for general data exploration. Two of these participants said that they would need more time to become proficient with the system before they would feel comfortable using it, but after that, they felt that they could use the system to gain more insights than they could from a 2D system alone. One participant in the AR condition said that they are a 3D learner which made interacting in this system easier for them because it fit their natural learning style. One participant in the AR group had aphantasia, a condition where people are unable to voluntary generate mental imagery. This

participant said that they ability to manipulate objects in 3D space felt better to them than a 2D system because it offloaded some of the need to generate a mental image of the data.

System Preference

Participants who had exposure to AR and VR were asked which device type they would prefer to use with this system. A participant (VR group) noted AR because they can become motion sick in VR. Another noted AR because they could move around the room more freely than they could with a VR system. The participant that chose VR did not want to use their hands to interact with the system, they were in the VR group. The other participants asked were undecided. Common sentiment was that AR would be better for in person collaboration but that in VR there were less distractions from the outside world, allowing increased immersion and focus.

4.6 Discussion

At the highest level, these results follow two significant themes. First, in the AR group, only half of the participants were able to successfully navigate the system while the other half faced a series of compounding difficulties. Second, there were notable differences in how participants interacted in the same environment between AR and VR use.

The interaction issues encountered by the bottom performers in the AR group may simply be the result of early usability testing on a novel system that used uncommon display technologies and interaction techniques. Early usability testing can be harmful or misleading if done to test new and unfamiliar interfaces [127]. The example given by Greenburg and Buxton (2008) was the early use of the radio where the process for setting it up and transmitting or receiving messages was onerous and likely to cause people to view the device as unusable [127].

This system is not the radio. The difficulties encountered with its use and its potential impact on society are both lesser. That said, this system was composed of pieces that are new to society. AR-HMDs and mid-air interactions are only recently becoming more common most prominently in industrial settings [34]. VR-HMDs have been more widely adopted by consumers, but most of these consumers are early tech adopters. Neither of these HMDs have widespread consumer use.

Another unfamiliar feature to users is the experience of viewing a 3D visualization in 3D space. This involves learning new interaction techniques and new ways to navigate the environment. The evaluation of this system was done early in these technology's life cycles.

Most participants were able to learn how to interact with this system, but many still faced difficulties. We believe that systems like this should be tested, but that the participants involved may not represent the average citizen. Participants may need to have exposure to stereoscopic displays and 3D interaction techniques before they can complete experimental tasks proficiently. Someday, most people will have exposure to the technologies used here, and until then, these systems should be tested carefully.

With the issues encountered during this early usability testing, we have taken the approach of not focusing on the correctness of interactions, questions, or answers. Instead, this work focuses on how interactions were performed and how the environment was navigated. These findings can help improve both this system and future systems by reducing barriers of use.

We believe that the combination of a complex environment, the optical see-through stereoscopic display, and mid-air gesture interactions were the main contributors to the struggles of the bottom performers of the AR group. Two of the top performers in the AR group had experience using VR-HMDs, meaning that they have used ray-cast interactions before. This base level of familiarity with ray-casting may have allowed these experienced users to focus more on the data rather than navigating the environment itself. The VR group had far fewer issues when interacting with the system.

4.6.1 AR/VR Display Differences

AR-HMDs and VR-HMDs use very different means of displaying virtual content. In this study, both devices used stereoscopic displays to render 3D content. When using the AR-HMD, participants could see the real world through the lenses that were displaying the virtual content whereas the VR-HMD blocked out the real world, showing instead a boundless virtual environment. These display differences inherently contribute to the ways that participants interact with the system.

Participants that could see the real world positioned the visualization on the desk in front of them and, on average, used a smaller visualization than the VR group. AR participants could also see themselves as well as the experimenter. In VR, participants could not see any of these things. In VR, participants could see where their controllers were, which when held, would show their hand positions but otherwise, visual cues for depth were missing.

Participants in VR, unable to see the desk, positioned the visualization in front of them based more on personal preference than real-world affordances. This manifested as participants placing the visualization further away from themselves and at different heights than AR participants. One VR participant placed the visualization partially under the desk, at one point causing their hand to hit the desk as they were interacting with an annotation. In addition to being further away, the VR group's visualizations were scaled to a larger size than the AR group's, and the only times the annotation station was placed at the participant's right was during the VR sessions.

We believe that the VR group felt more immersed in the environment and encountered fewer distractions from the real world. On the NASA TLX survey, VR group participants reported lower feelings of temporal demand and higher mental demand. This combination of scores can be interpreted as VR participants feeling more engaged in the environment, thus encountering more mental demand, but also being less impacted by perceived pressure to perform.

Conversely, we believe that the engagement of participants in the AR session, where the bottom performers nearly halted interactions with the system mid-way through the phases, was in part due to a loss of immersion. Instead of seeing a visual environment that consisted only of the data, these participants saw a researcher in a room. This perceived environment may have led to their increased feelings of time pressure and a limited willingness to fully interact with the virtual environment.

4.6.2 Interaction Technique Differences

Using the ray-cast projected line as a 3D cursor made sense in an environment where 3D objects existed at different distances from participants. Most, but not all, participants understood

this interaction technique. In both systems, the ray-cast representation and available use were the same; however, moving and selecting with the ray-cast required different actions.

The movement of the ray-cast was not notably different between the two devices. The ray-cast for each system had a one-to-one correspondence with the object it was projected from. In AR, ray-casts were projected from the center of a participants' hand away from their elbow. In VR, ray-casts were projected from the controllers. The ray-cast selection technique between the two systems was slightly more differentiated. Conceptually, both selection methods used a thumb press. In AR the thumb was pressed to the index finger and in VR the thumb was pressed against a button below it. It should be noted as a possible confounding factor that the pinch gesture in AR can be difficult for some people to execute. The user's hand must be angled slightly in towards the camera to enable device sensors to see the interaction. This adjustment of the hand does not move the ray-cast which is fixed more closely to a participants' wrist than their fingers.

Our leading hypothesis on the cause of the split between top performers and bottom performers in the AR group is that the difficulty they encountered when learning how to execute the pinch gesture triggered a series of compounding issues for the bottom performers. These bottom-performing participants had inconsistent successes with the pinch interaction causing them to feel more burden early in the experiment. This higher burden contributed to them using less effort when interacting with the system. In a sense, they became disengaged from the task. Their disengagement was further exacerbated when they were asked to produce questions and chose not to. Later, the difficulties with selection and limited experience with the system by users' choice, made finding the answers for questions during phase two difficult. These bottom performers took longer to complete the training sessions and reported higher feelings of time pressure while in the environment. The three top performers learned the selecting interaction more quickly which translated to shorter training times and higher performance with more time spent in the environment.

4.6.3 Participant Interaction Differences

Participants sat further away from the visualization in VR, making manipulation handles proportionally smaller and harder to target with a ray-cast. In AR, participants sat closer to the visualization causing the manipulation handles to be proportionally larger and easier to select. This difference may have contributed to VR participants standing rather than sitting and moving their heads more to view the data from different angles as opposed to AR participants which remained seated and used the manipulation handles to view the data. One participant in the VR group confirmed this after their session, saying that they chose not to rotate the visualization much because it was easier for them to move their head than for them to rotate the visualization.

With the visualization being further away, VR participants also more fully utilized their ray-casts by keeping their hands close to their bodies. In AR, participants kept their hands further out in front of them leading to higher reported fatigue.

The VR group was unencumbered by the real world, having a boundless virtual space to interact in. This lack of real-world references and some known issues with depth perception in VR [128] may have led to the differences in visualization placement and size. Issues with depth interpretation caused two participants in the VR group to request additional tools that could help indicate the depth of points in the scatter plot. One participant in VR even used a highlight annotation to check their ability to estimate depth in the environment by counting the points in an area and then using the highlight on the same area to confirm their count. Some VR participants encountered issues with this boundless space where they would move objects too far away from themselves to retrieve, a behavior that was not seen in AR. One VR participant requested that the environment be given virtual boundaries to help prevent this complication.

Instead of depth issues, AR participants struggled to see and interpret the colors used in the environment. When using the color mapping on the visualization, lighter-colored points became more difficult to see. This led one participant to use the centrality planes as a background to better contrast the colors of points.

4.6.4 Time in Environment

VR participants spent more time on average in the environment than the AR group. This was also true when comparing the VR group to the top AR performers and the bottom AR performers separately. VR participants completed the training faster than AR participants from both groups. These training times reflect the amount of time it took the participant to interact with each tool in the system, suggesting that VR participants picked up tools and features of this environment more quickly than AR participants.

In phase one, increased time in the environment was not associated with an increased number of questions asked. VR participants also spent more time in phase one and often interacted with the visualization and associated tools more than the AR group did. The VR group's increased exposure to the environment across both experiment phases likely improved their performance in phase two.

Increased time spent in the environment may also be related to the immersion that VR participants felt. They could not see the outside world, only the virtual environment, causing them to focus more on the tasks given. This additional mental effort is seen in the differences between the AR and VR NASA TLX scores. It is unclear why AR participants interacted with the system less, even among the participants that were skilled at using the ray-casts. It might also be that seeing the real world kept them from getting fully immersed in interactions with the environment.

4.6.5 Surveys Used

The three surveys used were selected because they measure spatial reasoning and graph literacy, both required components of this task; however, the results of these surveys did not provide a clear signal on participant performance. The participants that did poorly with the SGLS+ were not more likely to do perform poorly in this environment. Similarly, the highest and lowest scoring participants for the paper folding task both interacted comfortably in this environment, making a prediction of performance based on it difficult.

It is possible that high 2D graph literacy does not entirely transfer 3D graph literacy. The additional dimensions of data displayed may require a different form of graph literacy. In this environment, the visualization was directly manipulable. The direct interaction with 3D objects helped participants that self-reported low spatial reasoning (corroborated by the paper folding test results) to perform well in this environment. One participant commented that the level of direct interaction better fit their natural learning style. Another participant felt that interacting with the 3D visualization made understanding the data easier for them since they did not have to mentally compare different 2D graph states.

4.6.6 System Improvements

There are several improvements that could be made to this system based on these sessions and the participant's interview responses. In terms of tools, the addition of a slicing function would make viewing the scatter-plot easier for participants by allowing them to remove unneeded data. Trend lines were another commonly request tool that would make this environment more well-rounded for data analysis.

Improvements to Existing Tools

Details on Demand The details on-demand sphere was one of the most well-received tools; however, some modifications would improve it further. The three forms of information provided could be reduced to one which would reduce visual clutter in the system. The tick marks placed by the tool on the axes could also be removed. Most participants did not notice them, and the ones that did, didn't find much added value when using them.

The closest point sphere caused some confusion where participants were not sure which sphere to interact with. One participant suggested that the closet point sphere be removed and its details be displayed on the DoD sphere in place of the values associated with its current location. With this change, the sphere would alternate between showing actual data point values and relative position values. A different color or shape of text could be used to indicate which type of value was being shown. If this change was made, the data point that the values were being displayed for would

need to be highlighted. Otherwise, if several points are close together it would be difficult to tell which one had its values being displayed.

A different approach to implementing the DoD tool would be to attach it permanently to the user's hand or controller. That way they could always check values in the scatter-plot without needing to select the tool. While the DoD tool was one of the most used annotations, removing the need to generate and select it would streamline that part of interactions with the system.

Centrality Planes The centrality plane's major limitation were its buttons. They were difficult to interact with, especially at a distance. These buttons should be made larger to make them easier to press. They could also be removed from the plane so that they appear as a separate menu when the participant hovers over a marked area on the plane. This menu could be spawned with larger buttons where their size would not impact the visibility of the visualization.

Text A different text entry method would improve the functionality of the text box. This entry method would need to allow easier deletion of full words possibly by using a separate backspace button that removes a full word instead of a single character.

Highlight Volumes This system should be improved with a new highlight type that is a plane that can be fixed to one axis or moved freely. The plane would highlight all points on the value it was set to. This could look like highlighting all cereals with a fat content of 4 grams. This tool would improve the experience of users that had a difficult time determining where in the scatter-plot a point was located.

The existing highlights could also be improved with different scale and rotation methods. The manipulation handle method was difficult for participants to use. In place of handles, a series of sliders could be used to change the scale of each axis independently. These sliders could appear in a menu that loads when the highlight volume is selected. Another option would be to provide a rotation tool that could be used to rotate selected objects. Without the handles, the minimum size constraint could be reduced allowing users more freedom in how they use the highlight.

Environment Improvements

The addition of virtual boundaries and visual depth cues would improve the experience of VR users. These features would help VR users to better understand where they were placing the visualization and reduce the number of times they move objects too far away.

Adding a color selection tool for setting the color mapping would improve the experience of users by helping reduce the friction that poorly rendered colors in AR can cause. It would also allow participants to change the colors to match their preferences and visual ability, making the tool more inclusive to people with forms of colorblindness.

While not used during the pilots, speech controls for changing the axis mappings would be an improvement to the system. These commands would remove some of the direct interactions needed to change visualization states. The menu that participants used to set up the experiment and to level, center, or lock the visualization could be split into two menus; One menu for controlling the experiment setup and one for commonly used commands. The commonly used command menu would have the level, center, and lock buttons in addition to a few others. These new buttons would include one that removes all annotations from the visualization or all annotations of a certain type, with different buttons for the different types of annotations.

This system was designed such that annotations were linked to a combination of axis mappings, meaning annotations made when the axes were fat, sugar, and fiber would disappear when the axes were fat, sugar, and manufacturer. Annotations would reappear when the axis combination they were generated on was revisited. A new tool should be added that would serve as a persistent workspace. Annotations placed in that workspace would not be unloaded when the visualization changes. This workspace would allow participants to take notes that are preserved across graph states.

4.6.7 2D VS 3D Visualization Preference

Most participants enjoyed using this 3D system. They felt that it showed more data at once and was easier to draw conclusions from. The main disadvantage mentioned was that using 3D

systems was new and unfamiliar to participants who had spent years interacting with 2D visualizations. Participants acknowledged that using the 3D system would ultimately yield a more insightful experience, but that they would need to use the system over time and become more proficient with it before those benefits were realized.

4.6.8 System Preference

Not all participants had both AR and VR experience. The ones that did, agreed that AR would be better for co-located collaboration. It would allow people to see each other and the environment at the same time. VR was found to be better suited for individual data exploration and for remote collaboration. In VR, the outside world is removed which also removes many distractions from the system. VR IA systems used in corporate settings could leverage that ability to virtually remove office workers from their office making their workspaces feel larger than they are.

We believe that video pass-through headsets could provide the best of both worlds. When using one, the amount of the real-world to shown to users could be adjusted, allowing co-located collaboration and increased awareness of the user's surroundings when needed. That video of the real world could also be removed altogether, creating the immersive experience that VR-HMDs provide. Video pass-through also provides a richer color space and a wider field of view than can be displayed in AR, further improving the user's experience.

4.7 IA Experiment Design Guidelines

This section provides guidelines for other researchers working in this area. These guidelines are based on the experiences encountered over the development and execution of this experiment.

Introducing people to IA environments utilizing stereoscopic displays was difficult. It took participants a lot of training and interaction with the system before they were able to navigate it. Even by the end of the sessions, participants often commented that the system was unfamiliar to them. Most participants liked it but said they would need more time using it before they could feel comfortable in it. The difficulty of training participants was one of the reasons that the mid-air pen

tool was not used. The novelty of this system and the complexity of interacting in it can overwhelm participants, as seen with the bottom performers of the AR group.

We recommend that researchers plan on performing multiple sessions with participants. Ideally, there would be an initial training session where participants could become familiar with the environment and interact in an unstructured manner. Then later they could return to complete the experiment. This would allow more complex interactions to be tested. It would also enable participants to have higher performance at the task done during the experiment. If multiple sessions are not an option, recruiting for prior VR experience may be beneficial. That experience could help participants more quickly acclimate to the environment being used.

During the training phase, we recommend avoiding video instructions. The participants in this study gained a better understanding of the system when they were actively engaged and interacting with the system while training. Moreover, we recommend using a VR-HMD for training. The VR participants were more engaged with the system, interacting more with each tool and the visualization. This is seen in the lower training times and increased performance of the VR participants. Training participants on the system in VR can tap into that engagement and help reduce the difficulty of learning the system for new users.

4.8 Wizard of Oz Study Design

Our ambition of using this system to conduct a wizard of oz study was quickly diminished when pilot participants chose not to interact with the wizard-enabled features. The problem seemed to be twofold; learning the environment was complicated and the environment was partially interactive. If the environment was not interactive at all, participants would have needed to interact using the wizard. However, if this was the case, their interactions would be more forced and participants would likely be less engaged or less likely to believe that system was functioning.

Providing a training session beforehand on interacting with the system than training on the wizard enabled controls during the experiment could be another approach to resolving this issue. With that approach, the wizard capabilities of the system would be more salient in the participant's

mind and those participants would feel less overwhelmed with the amount of information provided at once. This technique needs further examination to assess its merit.

A benefit of a partially interactive IA WoZ interaction elicitation study design is that the interactions elicited can be more complex. Instead of needing participants to produce interactions for translations or other basic manipulations, they need only produce interactions for complex tasks such as changing an axis or highlighting a region of the data. This goal motivates us to continue investigating how to properly implement a semi-interactive WoZ study design.

4.9 Observations

Over the experimental sessions, several interesting themes in user behaviors were noted. Participants using the VR-HMD would typically use a larger visualization and place it without regard to the real-world, often placing it further away than the AR group. This resulted in VR wearers sitting further away from the visualization and interacting with it from a greater distance (Figure 4.7). In AR participants would usually place the visualization on the desk in front of them. Once placed, participants would interact with the visualization closely, often holding their hands near the visualization(Figure 4.7).

Apart from the visualizations scale and placement differences, there were differences in how VR users managed and navigated their virtual space. One such difference being that members of the VR group were the only ones who moved the annotation station from their left, where it was generated, to their right. With all participants being right-handed, it was interesting that only a few participants in the VR condition chose to move the most interacted with tool to their dominate side. Additionally and opposite to our original expectations, VR users were more likely to stand up and move around to view the data from different angles. These users did not walk around, but they did stand, lean, and move their upper body to see different views of the data. This was in contrast to the AR users who were more likely to rotate or move the visualization from where they were seated.

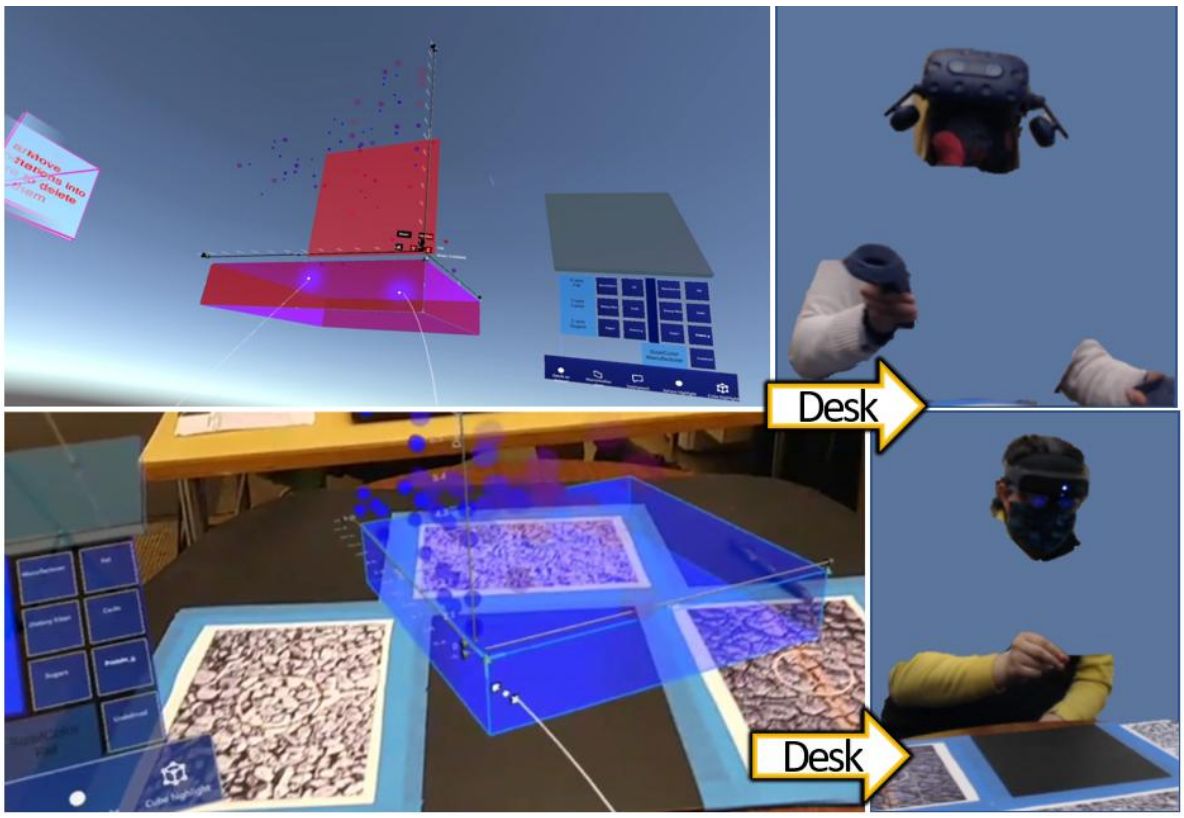


Figure 4.7: Top: VR group participant, Bottom: AR group participant. The VR participant is interacting with a larger visualization from a grater distance. Both participants are seated in-front of the same desk.

In either headset, participants were presented with a boundless virtual space, although in AR that space was physically limited by the size of the lab used. VR participants found the space to be too large, preferring instead that artificial boundaries be imposed upon that space. In contrast, AR participants wanted to have the option to scale the visualization to the size of the room, allowing them to walk through the data.

Depth perception was a separate issue faced by VR users, manifesting as a difficulties with counting points in a given area and identifying the location of a point in the visualization. A salient example of this was seen when one participant counted the points in a region of the scatter-plot then highlighted the same area to double-check their count against the highlights count. When asked about it, the participant explained the action by saying that they were checking their ability to accurately count the points before determining if they needed to use the highlight tool to answer questions.

AR participants had fewer issues with counting points, instead, they had difficulties with the colors rendered. Depending on the area the participant was looking at, the color/size mapping could make points more difficult to see. One participant use a mean/median plane as a quick fix to color perception issues by enlarging it and placing it as a backdrop behind the visualization, thus improving the color visibility.

4.10 Limitations

This work has several limitations. The first is the gender imbalance between the AR and VR groups. This gender imbalance may have impacted the results causing the differences to be between males and females and not between AR and VR. We do not believe this to be the case; however, more work is needed to tell the level of impact that the gender imbalance had on this study.

Another limitation faced was the difference between the ray-casts movement and selection across the two devices. We used the standard ray-cast techniques that come with the two devices used which may have led to differences between the two groups. The low performers in the AR group may have started the experiment in a disadvantaged state due to their difficulties with grasping the pinching gesture. It is also possible that the pinch gesture was not the issue and that those participants genuinely struggled to navigate the virtual 3D environment.

A major limitation faced by this work was the technological complexity and novelty of its design. The system developed and used here was complex and could be difficult to interact with. Many forms of feedback and affordances were provided by the system to help ease these interactions, but 3D stereoscopic IA environments are uncommon enough to be unfamiliar to most users. This observational study provides some of the groundwork needed to continue advancing research in IA, but it was limited by participants' difficulties with interaction in the system. This limitation was most salient with the 3 bottom performers in the AR group. This constraint suggests that researchers need to build more training into IA experiments or vary the platforms used for training

and testing. VR might be better for training than AR due to its removal of the distractions present in the real world, enabling participants to become more immersed in the environment.

4.11 Future Work

The SGLS and the additional scatter-plot questions asked were not strongly related to participant performance. This was also seen with the paper folding test. The sample size used in this study makes the survey's correlation with participant performance uncertain; however, we believe that IA researchers should develop a 3D graph literacy scale that can be completed in a stereoscopic HMD. It seems likely that 2D graph literacy and 3D graph literacy may be related but different skill-sets. This 3D graph literacy scale would be better suited for assessing people's ability to navigate 3D visualizations, as seen in IA environments.

This work would be improved by using a larger sample size. This study used a low sample size as justified by its aims at observing what participant behaviors are in the IA environment between the two tested devices. With its low sample size, it is difficult to tell if the three bottom performers in the AR group are an accurate reflection of how users would behave. Running this study on a larger sample size would provide more conclusive evidence as to the levels of performance that can be expected in AR IA environments.

The difference in the control of the ray-cast techniques used is another limitation of this work. A study that holds the ray-cast interactions constant would be more able to tell what participant interactions in IA are like without the confound of the controller used. Future work should implement more similar interaction control mechanisms to further tease apart the differences in interaction between AR and VR IA environments. One possible approach is to use a leap motion in conjunction with the VR-HMD to mirror the hand-based ray-cast used by the HoloLens 2. That said, this study was more able to tell how participants using the standard inputs of these emerging devices would interact, providing results that are more immediately applicable to systems using these devices as they are today.

4.12 Conclusion

This study compared interactions done in an IA environment between participants using an AR-HMD against ones using a VR-HMD and is one of the first studies in IA to compare interactions across these devices. There are two main findings of this work. First, not all participants in AR were able to interact successfully in the system, causing those participants to perform poorly and spend less time in the environment. These difficulties may have stemmed from struggles understanding how to select and navigate content in the environment. With those difficulties encountered early on, these AR participants became disengaged with the system, interacting less with it, and answering fewer questions about it.

Second, there were differences in how participants in AR compared to those in VR, navigated, interacted with, and arranged content. In VR participants were more immersed in the environment, leading to the increased time spent in the system, more interactions with the virtual content, and an increased ability to answer questions about the data presented. These VR participants also more fully utilized the space provided in the virtual environment, moving objects further away from themselves, and placing them with less concern for their position relative to the real world. AR participants spent longer in the training phase but less time in the other phases of the experiment. They all placed the visualization close to and in front of them on top of the real-world table. These participants interacted with the system less and reported higher fatigue from using mid-air gestures.

These differences in system use led us to recommend that future work consider using VR over AR when testing IA environments. This would allow participants to more fully interact with the task given during the experiment and make them more likely to be able to interact with the system. If that is not possible, or if testing the system in AR is required, we believe that multiple sessions should be conducted. These sessions would build the participant's ability to interact with the environment over time, mitigating the impacts that the new system, interaction techniques, and display technologies had on some participants in this study.

Chapter 5

Conclusion

This work stretched across a few domains and led to many findings. The main contributions of the earlier parts of this course of research are found in the interaction techniques and deeper user understanding that emerged from the two multimodal AR elicitation studies. Over these studies, gesture alone, speech alone, and gesture+speech interaction modalities were examined in an unconstrained AR environment.

Later, a complex and richly featured IA research platforms design and refinement are covered.

Finally, the differences between AR and VR IA system interaction and navigation are provided.

5.1 Contributions

This section summarizes the high-level contributions of the previous chapters.

5.1.1 Multimodal Interactions in Basic Augmented Reality Environments

Two multimodal elicitation studies were conducted during the first leg of this work (Chapter 2). These studies contribute useful interaction design guidelines and behavioral observations to the field. These observations cover the types of gestures used, the impact of referent display on elicitation results, the time differences between co-occurring gesture and speech use, the most common hand poses used when interacting in those environments, and a better understanding of how people interact in stereoscopic AR environments.

Major contributions of those two works include:

- A consensus gesture set for basic object manipulations in AR [29]
- A set of common hand poses used when making interactions in AR [30]
- Speech syntax use information [29, 30]
- Co-occurring gesture and speech time information [29, 30]

- The most common utterances used during speech interactions [30]
- Differences in perceived workload across the input modalities examined [29, 30]

5.1.2 Cross-platform Multi-user Immersive Analytics Platform

A major deliverable of this work is the interactive IA research system developed. This contribution includes the insights gained over the iterative development of the system and the reasoning for the design choices made. This system will be released as an open-source project pending the publication of this dissertation and it's associated publications. Once published, this system will be available at the natural user interaction lab website [12].

Major contributions of Chapter 3 were:

- A cross-platform multi-user immersive analytics platform that supports:
  - AR, VR, and cross-device use
  - Cross-platform environment coordinate system synchronization
  - Multi-user support
  - Co-located and remote collaboration
  - Asynchronous and synchronous collaboration
  - Wizard of Oz execution of interactions in the system
  - Data logging for all system events
  - Annotations and markup
  - Data analysis using provided tools
- Design guidelines based on iterative design sessions and experiences had while developing the system

[12] NUILAB.org

5.1.3 Augmented and Virtual Reality Immersive Analytics Interaction Comparison

Using the system covered in Chapter 3, an AR-VR IA interaction and use comparison study was run. That study details many of the differences found between using an AR-HMD for IA work compared to a VR-HMD. These comparisons were listed in Chapter 4. These comparisons provide a more complete picture of how users interact in AR and VR IA environments. While these findings represent early work, they provide valuable new information to the fields of IA and 3D user interaction design.

Major contributions of this chapter were:

- Differences found between AR and VR interactions in IA environments, including differences in:
  - Rotations performed
  - Physical movements performed
  - Translations performed
  - The use of virtual space
  - Participant ability to navigate the environment
  - Interaction technique use
  - Perceived workload when using this system
- Design guidelines for future work in IA
- Notes on and possible solutions to the issues encountered when setting up a semi-interactive WoZ study

5.2 Future Research Directions

This dissertation represents the early work done researching multimodal interactions in AR and VR environments. Many veins of research could follow up the studies presented here. Some of our

planned future work is to run a larger comparison between AR and VR IA use. This comparison will use more similar interaction techniques between the two devices. If possible, this comparison will use low latency video pass-through HMDs, allowing it to hold the device and the interaction techniques constant, while manipulating the amount of the real-world seen by the participant.

Implementing new and existing interaction techniques in this or other IA environments is another promising research path. In outlining how people choose to interact in AR environments when doing simple tasks and describing how using VR changes the way people interact inside of IA environments compared to AR, this work provides the foundation necessary to start building intuitive and effective interaction techniques for basic and complex AR IA environments. Results of these studies will be available to help researchers determine which interaction techniques are transferable Between VR and AR environments. Improving these interaction techniques will increase novice users' ability to interact with virtual systems, diminishing the interaction difficulties observed during this work.

In addition to improving the interaction techniques used, this system itself could continue development, starting with the improvements consequent of the AR-VR comparison study. Incorporating these will increasingly ready this platform for ongoing research.

A goal that this work was unable to meet was the execution of a semi-interactive WoZ study within a complex environment. We believe that conducting two sessions with participants, one for training, and one for the experimental task, would allow that style of study to be run; however, more work is needed to determine whether that course of action is enough to solve the issues encountered here.

5.3 Limitations

As with most research, this work was met with many limitations. The first studies run were limited by their simplicity, resulting in findings that were most applicable to less complex domains. Those works were also restricted by the difficulties encountered while performing multimodal elicitation studies due to the ways that the referents primed some of the results found in each study.

The IA system designed and used during this research was limited by the technologies used and the complexity of the environment. Designing a system that connects several cutting-edge devices across various code bases was difficult. The impact of these limitations is evident in the system's current ability to only support scatter-plot visualizations while the IATK supports several other graph types [1]. Another salient limitation of this system is that when testing collaborative tasks some users can experience latency. This latency is typically minimal but is relevant if both parties are co-located.

The gender imbalance between the AR and VR groups, the differences in ray-cast movement and selection, the novelty of the devices used, and the complexity of the experimental task each posed constraints during the AR-VR comparison study. The gender imbalance may have biased the results of one, or both devices compared. The novelty of the environment displayed, devices used and their associated interaction techniques all contributed to the struggles of the bottom performers in the AR group. These participants may or may not be an accurate representation of how other users would perform in this system. The combination of more research in this area improved IA training and further evolved IA platforms will help reduce these limitations for future work.

5.4 Final Remarks

Stereoscopic displays can provide virtual experiences that are not possible when using a traditional desktop computer, experiences including the IA environment used here, the IA systems found in other works [1, 118, 129], surgical training platforms [130], immersive video gaming [13], and 3D immersive movies [14]. Proper implementation of these experiences can increase engagement with a system [10, Chapter 6] or deepen the understanding of what is being taught [20]. Some of the experiences afforded by these devices may not require a robust interaction system (e.g., watching a movie), however; many of these virtual experiences will need to allow users to comfortably interact before their full benefit can be realized.

[13] https://www.oculus.com/

[14] https://cinevr.io/en

When viewing 3D visualizations, users will need to feel able to navigate the environment. Without careful attention to proper interaction technique design, the struggles encountered by half of the AR group may occur in other novice users. These struggles may range from a system that is uncomfortable and not fit for long-term use, or they could be as major as causing new users to not being able to interact in these environments at all.

Getting the interactions right for AR and VR environments is a step towards facilitating their widespread acceptance. This dissertation provides the groundwork needed to start designing interaction techniques around how people utilize their personal space, virtual space, body, IA tools, and feedback systems.

Bibliography

[1] M. Cordeil, A. Cunningham, B. Bach, C. Hurter, B. H. Thomas, K. Marriott, and T. Dwyer. Iatk: An immersive analytics toolkit. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 200–209, 2019.

[2] R Ekstrome, J French, H Harman, and D Dermen. Manual for kit of factor referenced cognitive tests. *Princeton, New Jersey: Educational Testing Service*, 1976, 1976.

[3] Joshua Brustein. Microsoft wins $480 million army battlefield contract, Nov 2018.

[4] B. Lee, E. K. Choe, P. Isenberg, K. Marriott, and J. Stasko. Reaching broader audiences with data visualization. *IEEE Computer Graphics and Applications*, 40(2):82–90, March 2020.

[5] Melvin M Vopson. The information catastrophe. *AIP Advances*, 10(8):085014, 2020.

[6] IBM Cloud. Ibm cloud pak for data, Aug 2020.

[7] William Buxton and Brad Myers. A study in two-handed input. *ACM SIGCHI Bulletin*, 17(4):321–326, 1986.

[8] 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*, CHI '09, pages 1083–1092, New York, NY, USA, 2009. ACM.

[9] Minkyung Lee and Mark Billinghurst. A wizard of oz study for an ar multimodal interface. In *Proceedings of the 10th International Conference on Multimodal Interfaces*, ICMI '08, page 249–256, New York, NY, USA, 2008. Association for Computing Machinery.

[10] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H Thomas. *Immersive Analytics*, volume 11190. Springer, 2018.

[11] A. Fonnet and Y. Prié. Survey of immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2019.

[12] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pages 1–8, 2015.

[13] Emily Courtney. 27 companies that have switched to long-term remote work, Nov 2020.

[14] Tom Warren. Microsoft is letting more employees work from home permanently, Oct 2020.

[15] Rob McLean. These companies plan to make working from home the new normal. as in forever, Jun 2020.

[16] Stefan Marks, Javier E. Estevez, and Andy M. Connor. Towards the holodeck: Fully immersive virtual reality visualisation of scientific and engineering data. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, IVCNZ '14, page 42–47, New York, NY, USA, 2014. Association for Computing Machinery.

[17] Zhuming Ai and Torsten Fröhlich. Molecular dynamics simulation in virtual environments. In *Computer Graphics Forum*, number 3 in 17, pages 267–273. Wiley Online Library, 1998.

[18] S. Zhang, C. Demiralp, D. F. Keefe, M. DaSilva, D. H. Laidlaw, B. D. Greenberg, P. J. Basser, C. Pierpaoli, E. A. Chiocca, and T. S. Deisboeck. An immersive virtual environment for dt-mri volume visualization applications: a case study. In *Proceedings Visualization, 2001. VIS '01.*, pages 437–584, 2001.

[19] Sebastian Blum, Gokhan Cetin, and Wolfgang Stuerzlinger. Immersive analytics sensemaking on different platforms. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 01 2019.

[20] Yea-Seul Kim, Nathalie Henry Riche, Bongshin Lee, Matthew Brehmer, Michel Pahud, Ken Hinckley, and Jessica Hullman. Inking your insights: Investigating digital externalization

behaviors during data analysis. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ISS '19, page 255–267, New York, NY, USA, 2019. Association for Computing Machinery.

[21] Yedendra Babu Shrinivasan and Jarke J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1237–1246, New York, NY, USA, 2008. Association for Computing Machinery.

[22] Seyoung Park and Donghee Shin. Effects of text input system on learner's memory: Handwriting versus typing on tablet pc. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*, IMCOM '15, New York, NY, USA, 2015. Association for Computing Machinery.

[23] Yann Riche, Nathalie Henry Riche, Ken Hinckley, Sheri Panabaker, Sarah Fuelling, and Sarah Williams. As we may ink? learning from everyday analog pen use to improve digital ink experiences. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3241–3253, New York, NY, USA, 2017. Association for Computing Machinery.

[24] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. Pen + touch = new tools. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, page 27–36, New York, NY, USA, 2010. Association for Computing Machinery.

[25] Ken Pfeuffer, Ken Hinckley, Michel Pahud, and Bill Buxton. Thumb + pen interaction on tablets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3254–3266, New York, NY, USA, 2017. Association for Computing Machinery.

[26] A. U. Batmaz, A. K. Mutasim, and W. Stuerzlinger. Precision vs. power grip: A comparison of pen grip styles for selection in virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 23–28, 2020.

[27] Travis Gesslein, Verena Biener, Philipp Gagel, Daniel Schneider, Per Ola Kristensson, Eyal Ofek, Michel Pahud, and Jens Grubert. Pen-based interaction with spreadsheets in mobile virtual reality. *arXiv preprint arXiv:2008.04543*, 2020.

[28] Poorna Talkad Sukumar, Anqing Liu, and Ronald Metoyer. Replicating user-defined gestures for text editing. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, ISS '18, page 97–106, New York, NY, USA, 2018. Association for Computing Machinery.

[29] Adam S. Williams, Jason Garcia, and Francisco Ortega. Understanding multimodal user gesture and speech behavior for object manipulation in augmented reality using elicitation. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3479–3489, 2020. ©2020 IEEE. Reprinted, with permission from Adam S. Williams, Jason Garcia, and Francisco Ortega, Paper Title: Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation, IEEE publication title: Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation, December 2020.

[30] Adam S. Williams and Francisco R. Ortega. Understanding gesture and speech multimodal interactions for manipulation tasks in augmented reality using unconstrained elicitation. *Proc. ACM Hum.-Comput. Interact.*, 4(ISS), nov 2020. ©2020 ACM. Reprinted, with permission from Adam S. Williams, Jason Garcia, and Francisco Ortega.

[31] Adam S. Williams, Sarah Coler, and Francisco Ortega. Conversations on multimodal input design with older adults, 2020. https://arxiv.org/abs/2008.11834.

[32] Adam S. Williams and Francisco R. Ortega. A concise guide to elicitation methodology, 2021. https://arxiv.org/abs/2105.12865.

[33] Francisco R. Ortega, Adam Williams, and Jason Garcia. Multi-modal gesture elicitation methodology for children. In *Proceedings of the 2020 ACM Interaction Design and Children Conference: Extended Abstracts*, IDC '20, page 85–88, New York, NY, USA, 2020. Association for Computing Machinery.

[34] Council post: 10 intriguing uses of ar technology in industry, Jul 2020. https://www.forbes.com/sites/forbestechcouncil/2020/07/14/10-intriguing-uses-of-ar-technology-in-industry.

[35] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland, USA, 1988.

[36] Sabrina Connell, Pei-Yi Kuo, Liu Liu, and Anne Marie Piper. A wizard-of-oz elicitation study examining child-defined gestures with a whole-body interface. In *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, page 277–280, New York, NY, USA, 2013. Association for Computing Machinery.

[37] F. R. Ortega, A. Galvan, K. Tarre, A. Barreto, N. Rishe, J. Bernal, R. Balcazar, and J. Thomas. Gesture elicitation for 3d travel via multi-touch and mid-air systems for procedurally generated pseudo-universe. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 144–153, Los Angeles, CA, USA, 2017. IEEE.

[38] Jaime Ruiz and Daniel Vogel. Soft-constraints to reduce legacy and performance bias to elicit whole-body gestures with low arm fatigue. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 3347–3350, New York, NY, USA, 2015. Association for Computing Machinery.

[39] Ionuţ-Alexandru Zaiţi, Ştefan-Gheorghe Pentiuc, and Radu-Daniel Vatavu. On free-hand tv control: experimental results on user-elicited gestures with leap motion. *Personal and Ubiquitous Computing*, 19(5-6):821–838, 2015.

[40] Radu-Daniel Vatavu. There's a world outside your tv: Exploring interactions beyond the physical tv screen. In *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV '13, page 143–152, New York, NY, USA, 2013. Association for Computing Machinery.

[41] Nem Khan Dim, Chaklam Silpasuwanchai, Sayan Sarcar, and Xiangshi Ren. Designing mid-air tv gestures for blind people using user- and choice-based elicitation approaches. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, DIS '16, page 204–214, New York, NY, USA, 2016. Association for Computing Machinery.

[42] Lynn Hoff, Eva Hornecker, and Sven Bertel. Modifying gesture elicitation: Do kinaesthetic priming and increased production reduce legacy bias? In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, page 86–91, New York, NY, USA, 2016. Association for Computing Machinery.

[43] Sumbul Khan and Bige Tunçer. Gesture and speech elicitation for 3d cad modeling in conceptual design. *Automation in Construction*, 106:102847, 2019.

[44] Keenan R. May, Thomas M. Gable, and Bruce N. Walker. Designing an in-vehicle air gesture set using elicitation methods. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '17, page 74–83, New York, NY, USA, 2017. Association for Computing Machinery.

[45] Panayiotis Koutsabasis and Chris K. Domouzis. Mid-air browsing and selection in image collections. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, page 21–27, New York, NY, USA, 2016. Association for Computing Machinery.

[46] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., USA, 2004.

[47] Francisco R Ortega, Fatemeh Abyarjoo, Armando Barreto, Naphtali Rishe, and Malek Adjouadi. *Interaction design for 3D user interfaces: The world of modern input devices for research, applications, and game development*. CRC Press, 2016.

[48] Santiago Villarreal-Narvaez, Jean Vanderdonckt, Radu-Daniel Vatavu, and Jacob A Wobbrock. A systematic review of gesture elicitation studies: What can we learn from 216 studies. In *Proceedings of ACM Int. Conf. on Designing Interactive Systems (DIS'20)*, page NA, Eindhoven, 2020. ACM Press.

[49] Radu-Daniel Vatavu and Jacob O. Wobbrock. Formalizing agreement analysis for elicitation studies: New measures, significance test, and toolkit. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1325–1334, New York, NY, USA, 2015. Association for Computing Machinery.

[50] Theophanis Tsandilas. Fallacies of agreement: A critical review of consensus assessment methods for gesture elicitation. *ACM Trans. Comput. Hum. Interact.*, 25(3):18, June 2018.

[51] Meredith Ringel Morris. Web on the wall: Insights from a multimodal interaction elicitation study. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '12, pages 95–104, New York, NY, USA, 2012. ACM.

[52] Andreas Dünser, Raphaël Grasset, Hartmut Seichter, and Mark Billinghurst. *Applying HCI principles to AR systems design*. University of Canterbury. Human Interface Technology Laboratory., New Zealand, 2007.

[53] David Mcneill. *Gesture and Thought*. the University of Chicago Press, USA, 01 2005.

[54] Spencer D Kelly, Asli Ozyürek, and Eric Maris. Two sides of the same coin: speech and gesture mutually interact to enhance comprehension. *Psychol. Sci.*, 21(2):260–267, February 2010.

[55] Lisette Mol and Sotaro Kita. Gesture structure affects syntactic structure in speech. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 34, pages 761 – 766, USA, 2012. CogSci.

[56] Andrea Corradini and Philip R Cohen. On the relationships among speech, gestures, and object manipulation in virtual environments: Initial evidence. In *Advances in Natural Multimodal Dialogue Systems*, pages 97–112. Springer, 2005.

[57] Sébastien Carbini, Lionel Delphin-Poulat, L Perron, and Jean-Emmanuel Viallet. From a wizard of oz experiment to a real time speech and gesture multimodal interface. *Signal Processing*, 86(12):3559–3577, 2006.

[58] Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum. A procedure for developing intuitive and ergonomic gesture interfaces for hci. In Antonio Camurri and Gualtiero Volpe, editors, *Gesture-Based Communication in Human-Computer Interaction*, pages 409–420, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[59] Richard A. Bolt. “put-that-there”: Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 14(3):262–270, July 1980.

[60] Andrea Corradini and Philip R Cohen. On the relationships among speech, gestures, and object manipulation in virtual environments: Initial evidence, 2005.

[61] Minkyung Lee, Mark Billinghurst, Woonhyuk Baek, Richard Green, and Woontack Woo. A usability study of multimodal input in an augmented reality environment. *Virtual Real.*, 17(4):293–305, November 2013.

[62] S Goldin-Meadow, H Nusbaum, S D Kelly, and S Wagner. Explaining math: gesturing lightens the load. *Psychol. Sci.*, 12(6):516–522, November 2001.

[63] Susan Goldin-Meadow, Martha Wagner Alibali, and R Breckinridge Church. Transitions in concept acquisition: using the hand to read the mind. *Psychological review*, 100(2):279, 1993.

[64] Sharon Oviatt. Taming recognition errors with a multimodal interface. *Communications of the ACM*, 43(9):45–51, 2000.

[65] David B. Koons, Carlton J. Sparrell, and Kristinn Rr. Thorisson. *Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures*, page 53–64. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[66] Ed Kaiser, Alex Olwal, David McGee, Hrvoje Benko, Andrea Corradini, Xiaoguang Li, Phil Cohen, and Steven Feiner. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *Proceedings of the 5th International Conference on Multimodal Interfaces*, ICMI '03, page 12–19, New York, NY, USA, 2003. Association for Computing Machinery.

[67] A G Hauptmann. Speech and gestures for graphic image manipulation. *ACM SIGCHI Bulletin*, 20(SI):241–245, 1989.

[68] Joyce Y. Chai and Shaolin Qu. A salience driven approach to robust input interpretation in multimodal conversational systems. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, page 217–224, USA, 2005. Association for Computational Linguistics.

[69] A A Karpov and R M Yusupov. Multimodal interfaces of Human–Computer interaction. *Her. Russ. Acad. Sci.*, 88(1):67–74, January 2018.

[70] Muhammad Zeeshan Baig and Manolya Kavakli. Qualitative analysis of a multimodal interface system using speech/gesture. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 2811–2816, Wuhan, China, 2018. IEEE, IEEE.

[71] Katrin Wolf, Anja Naumann, Michael Rohs, and Jörg Müller. Taxonomy of microinteractions: Defining microgestures based on ergonomic and scenario-dependent requirements. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interac-*

tion - Volume Part I, INTERACT'11, pages 559–575, Berlin, Heidelberg, 2011. Springer-Verlag.

[72] Miguel A Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. Memorability of pre-designed and user-defined gesture sets, 2013.

[73] Katherine Tarre, Adam S. Williams, Lukas Borges, Naphtali D. Rishe, Armando B. Barreto, and Francisco R. Ortega. Towards first person gamer modeling and the problem with game classification in user studies. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, pages 125:1–125:2, New York, NY, USA, 2018. ACM.

[74] Thomas Plank, Hans-Christian Jetter, Roman Rädle, Clemens N. Klokmose, Thomas Luger, and Harald Reiterer. Is two enough?: ! studying benefits, barriers, and biases of multi-tablet use for collaborative visualization. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 4548–4560, New York, NY, USA, 2017. ACM.

[75] F. R. Ortega, K. Tarre, M. Kress, A. S. Williams, A. B. Barreto, and N. D. Rishe. Selection and manipulation whole-body gesture elicitation study in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1723–1728, Osaka, Japan, Japan, 2019. IEEE.

[76] Jacob O Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A Myers. Maximizing the guessability of symbolic input, 2005.

[77] Radu-Daniel Vatavu and Jacob O. Wobbrock. Between-subjects elicitation studies: Formalization and tool support. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3390–3402, New York, NY, USA, 2016. Association for Computing Machinery.

[78] Panagiotis Vogiatzidakis and Panayiotis Koutsabasis. Gesture elicitation studies for Mid-Air interaction: A review. *Multimodal Technologies and Interaction*, 2(4):65, September 2018.

[79] Aurélie Cohé and Martin Hachet. Understanding user gestures for manipulating 3d objects from touchscreen inputs. In *Proceedings of Graphics Interface 2012*, GI '12, pages 157–164, Toronto, Ont., Canada, Canada, 2012. Canadian Information Processing Society.

[80] Sarah Buchanan, Bourke Floyd, Will Holderness, and Joseph J. LaViola. Towards user-defined multi-touch gestures for 3d objects. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS '13, page 231–240, New York, NY, USA, 2013. Association for Computing Machinery.

[81] Mark Micire, Munjal Desai, Amanda Courtemanche, Katherine M. Tsui, and Holly A. Yanco. Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 41–48, New York, NY, USA, 2009. ACM.

[82] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction, 2011.

[83] Ionuţ-Alexandru Zaiţi, Ştefan-Gheorghe Pentiuc, and Radu-Daniel Vatavu. On free-hand TV control: experimental results on user-elicited gestures with leap motion. *Pers. Ubiquit. Comput.*, 19(5):821–838, August 2015.

[84] Markus L Wittorf and Mikkel R Jakobsen. Eliciting Mid-Air gestures for Wall-Display interaction. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, NordiCHI '16, pages 3:1–3:4, New York, NY, USA, 2016. ACM.

[85] Aakar Gupta, Thomas Pietrzak, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. Summon and select: Rapid interaction with interface controls in mid-air. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, pages 52–61, New York, NY, USA, 2017. ACM.

[86] Bashar Altakrouri, Daniel Burmeister, Dennis Boldt, and Andreas Schrader. Insights on the impact of physical impairments in full-body motion gesture elicitation studies. In *Proceed-*

ings of the 9th Nordic Conference on Human-Computer Interaction, NordiCHI '16, pages 5:1–5:10, New York, NY, USA, 2016. ACM.

[87] Alexander G Hauptmann and Paul McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38(2):231–249, 1993.

[88] Christophe Mignot, Claude Valot, and Noëlle Carbonell. An experimental study of future "natural" multimodal human-computer interaction. In *INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems*, CHI '93, page 67–68, New York, NY, USA, 1993. Association for Computing Machinery.

[89] Sandrine Robbe. An empirical study of speech and gesture interaction: Toward the definition of ergonomic design guidelines. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, page 349–350, New York, NY, USA, 1998. Association for Computing Machinery.

[90] Dimitra Anastasiou, Cui Jian, and Desislava Zhekova. Speech and gesture interaction in an ambient assisted living lab. In *Proceedings of the 1st Workshop on Speech and Multimodal Interaction in Assistive Environments*, SMIAE '12, pages 18–27, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[91] Sylvia Irawati, Scott Green, Mark Billinghurst, Andreas Duenser, and Heedong Ko. An evaluation of an augmented reality multimodal interface using speech and paddle gestures. In *Proceedings of the 16th International Conference on Advances in Artificial Reality and Tele-Existence*, ICAT'06, page 272–283, Berlin, Heidelberg, 2006. Springer-Verlag.

[92] Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pittman, and Ira Smith. Unification-based multimodal integration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98/EACL '98, page 281–288, USA, 1997. Association for Computational Linguistics.

[93] Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, page 415–422, New York, NY, USA, 1997. Association for Computing Machinery.

[94] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. User elicitation on single-hand microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3403–3414, New York, NY, USA, 2016. Association for Computing Machinery.

[95] Marie-Luce Bourguet and Akio Ando. Synchronization of speech and hand gestures during multimodal human-computer interaction. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, page 241–242, New York, NY, USA, 1998. Association for Computing Machinery.

[96] Susan Wagner Cook and Michael K Tanenhaus. Embodied communication: Speakers' gestures affect listeners' actions. *Cognition*, 113(1):98–104, 2009.

[97] Anne Köpsel and Nikola Bubalo. Benefiting from legacy bias. *interactions*, 22(5):44–47, August 2015.

[98] Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, M c Schraefel, and Jacob O Wobbrock. Reducing legacy bias in gesture elicitation studies. *Interactions*, 21(3):40–45, May 2014.

[99] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, page 955–960, New York, NY, USA, 2013. Association for Computing Machinery.

[100] Daniel P Loehr. Temporal, structural, and pragmatic synchrony between intonation and gesture. *Laboratory Phonology*, 3(1):71–89, 2012.

[101] Emanuel A Schegloff. On some gestures' relation to talk.(pp. 266-296) in j. maxwell and j. heritage (eds.) structures of social action, 1984.

[102] Alexander G. Hauptmann and Paul McAvinney. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies*, 38(2):231 – 249, 1993.

[103] 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*, pages 1275–1284. ACM, 2008.

[104] Kenrick Kin, Maneesh Agrawala, and Tony DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proceedings of Graphics interface 2009*, pages 119–124. Canadian Information Processing Society, 2009.

[105] Meredith Ringel Morris, Jacob O Wobbrock, and Andrew D Wilson. Understanding users' preferences for surface gestures. In *Proceedings of graphics interface 2010*, pages 261–268. Canadian Information Processing Society, 2010.

[106] Sharon Oviatt. Mulitmodal interactive maps: Designing for human performance. *Human–Computer Interaction*, 12(1-2):93–129, 1997.

[107] T. Piumsomboon, D. Altimira, H. Kim, A. Clark, G. Lee, and M. Billinghurst. Grasp-shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 73–82, Munich, Germany, 2014. IEEE.

[108] Susumu Harada, Daisuke Sato, Hironobu Takagi, and Chieko Asakawa. Characteristics of elderly user behavior on mobile multi-touch devices. In Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2013*, pages 323–341, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[109] Joshua Brustein. Microsoft wins $480 million army battlefield contract, Nov 2018.

[110] Helge Petersson, David Sinkvist, Chunliang Wang, and Örjan Smedby. Web-based interactive 3d visualization as a tool for improved anatomy learning. *Anatomical sciences education*, 2(2):61–68, 2009.

[111] Robert J.K. Jacob, Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: A framework for post-wimp interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 201–210, New York, NY, USA, 2008. Association for Computing Machinery.

[112] Sharon Oviatt, Rachel Coulston, and Rebecca Lunsford. When do we interact multimodally? cognitive load and multimodal communication patterns. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, page 129–136, New York, NY, USA, 2004. Association for Computing Machinery.

[113] Sharon Oviatt. Ten myths of multimodal interaction. *Commun. ACM*, 42(11):74–81, November 1999.

[114] Marie-Luce Bourguet. Towards a taxonomy of error-handling strategies in recognition-based multi-modal human–computer interfaces. *Signal Processing*, 86(12):3625–3643, 2006.

[115] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Predicting the cost of error correction in character-based text entry technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 5–14, New York, NY, USA, 2010. Association for Computing Machinery.

[116] Ohoud Alharbi, Ahmed Sabbir Arif, Wolfgang Stuerzlinger, Mark D. Dunlop, and Andreas Komninos. Wisetype: A tablet keyboard with color-coded visualization and various editing options for error correction. In *Proceedings of Graphics Interface 2019*, GI 2019. Canadian Information Processing Society, 2019.

[117] Yi-Jheng Huang, Takanori Fujiwara, Yun-Xuan Lin, Wen-Chieh Lin, and Kwan-Liu Ma. A gesture system for graph visualization in virtual reality environments. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 41–45. IEEE, 2017.

[118] R. Sicat, J. Li, J. Choi, M. Cordeil, W. Jeong, B. Bach, and H. Pfister. Dxr: A toolkit for building immersive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):715–725, 2019.

[119] Benjamin Lee, Dave Brown, Bongshin Lee, Christophe Hurter, Steven Drucker, and Tim Dwyer. Data visceralization: Enabling deeper understanding of data using virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1095–1105, 2021.

[120] Mayra Donaji Barrera Machuca, Wolfgang Stuerzlinger, and Paul Asente. The effect of spatial ability on immersive 3d drawing. In *Proceedings of the 2019 on Creativity and Cognition*, C&C '19, page 173–186, New York, NY, USA, 2019. Association for Computing Machinery.

[121] Philipp Eichmann, Darren Edge, Nathan Evans, Bongshin Lee, Matthew Brehmer, and Christopher White. Orchard: Exploring multivariate heterogeneous networks on mobile phones. In *Computer Graphics Forum*, volume 39, pages 115–126. Wiley Online Library, 2020.

[122] Yasmina Okan, Eva Janssen, Mirta Galesic, and Erika A Waters. Using the short graph literacy scale to predict precursors of health behavior change. *Medical Decision Making*, 39(3):183–195, 2019.

[123] Mirta Galesic and Rocio Garcia-Retamero. Graph literacy: a cross-cultural comparison. *Medical Decision Making*, 31(3):444–457, 2011.

[124] James Ainooson and Maithilee Kunda. A computational model for reasoning about the paper folding task using visual mental images. In *CogSci*, 2017.

[125] Wallace S. Lages and Doug A. Bowman. Move the object or move myself? walking vs. manipulation for the examination of 3d scientific data. *Frontiers in ICT*, 5:15, 2018.

[126] Jorge Wagner, Wolfgang Stuerzlinger, and Luciana Nedel. The effect of exploration mode and frame of reference in immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2021.

[127] Saul Greenberg and Bill Buxton. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 111–120, New York, NY, USA, 2008. Association for Computing Machinery.

[128] Fatima El Jamiy and Ronald Marsh. Distance estimation in virtual reality and augmented reality: A survey. In *2019 IEEE International Conference on Electro Information Technology (EIT)*, pages 063–068, 2019.

[129] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. Imaxes: Immersive axes as embodied affordances for interactive multivariate data visualisation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, page 71–83, New York, NY, USA, 2017. Association for Computing Machinery.

[130] Chi Jin, Liuyan Dai, and Tong Wang. The application of virtual reality in the training of laparoscopic surgery: A systematic review and meta-analysis. *International Journal of Surgery*, 87:105859, 2021.

[131] Adam S. Williams and Francisco R. Ortega. Using a 6 degrees of freedom virtual reality input device with an augmented reality headset in a collaborative environment. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 205–209, 2021. ©2021 IEEE. Reprinted, with permission from Adam S. Williams and Francisco Ortega, Paper Title: Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment, IEEE publication

title: Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment, May 2021.

[132] Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. Explosion diagrams in augmented reality. In *2009 IEEE Virtual Reality Conference*, pages 71–78. IEEE, 2009.

[133] Adam Sinclair Williams, Catherine Angelini, Mathew Kress, Edgar Ramos Vieira, Newton D'Souza, Naphtali D. Rishe, Joseph Medina, Ebru Özer, and Francisco Ortega. Augmented reality for city planning. In Jessie Y. C. Chen and Gino Fragomeni, editors, *Virtual, Augmented and Mixed Reality. Design and Interaction*, pages 256–271, Cham, 2020. Springer International Publishing.

[134] A. S. Williams and F. Ortega. Insights on visual aid and study design for gesture interaction in limited sensor range augmented reality devices. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 19–22, 2020.

[135] Michelle A. Brown and Wolfgang Stuerzlinger. Exploring the throughput potential of in-air pointing. In Masaaki Kurosu, editor, *Human-Computer Interaction. Interaction Platforms and Techniques*, pages 13–24, Cham, 2016. Springer International Publishing.

[136] Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. Arpen: Mid-air object manipulation techniques for a bimanual ar system with pen & smartphone. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery.

[137] Duc-Minh Pham and Wolfgang Stuerzlinger. Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, VRST '19, New York, NY, USA, 2019. Association for Computing Machinery.

[138] Junwei Sun, Wolfgang Stuerzlinger, and Bernhard E. Riecke. Comparing input methods and cursors for 3d positioning with head-mounted displays. In *Proceedings of the 15th*

ACM Symposium on Applied Perception, SAP '18, New York, NY, USA, 2018. Association for Computing Machinery.

[139] Huidong Bai, Lei Gao, and Mark Billinghurst. 6dof input for hololens using vive controller. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, SA '17, New York, NY, USA, 2017. Association for Computing Machinery.

[140] Felix Kosmalla, André Zenner, Marco Speicher, Florian Daiber, Nico Herbig, and Antonio Krüger. *Exploring Rock Climbing in Mixed Reality Environments*, page 1787–1793. Association for Computing Machinery, New York, NY, USA, 2017.

[141] R. J. Teather, A. Pavlovych, W. Stuerzlinger, and I. S. MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *2009 IEEE Symposium on 3D User Interfaces*, pages 43–50, 2009.

[142] Installation guide | mixed reality toolkit documentation, Jan 2020. Available at https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Installation.html#4-add-and-configure-mrtk-with-a-new-scene.

[143] Lauren Goode. The hololens 2 puts a full-fledged computer on your face, Feb 2019. Available at https://www.wired.com/story/microsoft-hololens-2-headset/.

Appendix A

Appendix

A.1 Other Works Done During This Degree

Refereed Journals

- **Williams, A. S.**, Garcia, J., De Zayas, F., Hernandez, F. Sharp, J., and Ortega, F. (2020). "The Cost of Production in Elicitation Studies and the Legacy Bias-Consensus Trade off". *Multimodal Technologies and Interaction, 4, 88. DOI: https://doi.org/10.3390/mti4040088*

- **Williams, A. S.**, *Ortega, F. (2020). "Understanding Gesture and Speech Multimodal Interactions for Manipulation Tasks in Augmented Reality Using Unconstrained Elicitation". Proc. ACM Human-Computer Interaction. V4, ISS, Article 202 (November 2020), 21 pages. DOI: https://doi.org/10.1145/3427330*

- **Williams, A. S.**, *Garcia, J., Ortega, F. (2020). "Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation". in IEEE Transactions on Visualization and Computer Graphics, DOI: https://doi.org/10.1109/TVCG.2020.3023566, Impact Factor: 4.56, Acceptance Rate: 6%*

Refereed Workshop Articles

- **Williams, A**., Ortega, F. (2021), "Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment". In *IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*,2021, pp. 205-209, DOI: https://doi:10.1109/VRW52623.2021.00045

- **Williams, A**., Ortega, F. (2020), "Multimodal User-Defined inputs for Optical See Through Augmented Reality Environments". In *IEEE Conference on Virtual Reality and 3D User*

Interfaces Abstracts and Workshops (VRW), pp. 557-558, DOI: https://doi.org/10.1109/VRW50115.2020.00130

- **Williams, A.S**., Ortega, F. (2020), "Conversations On Multimodal Input Design With Older Adults," *CHI 2020 (Designing Interactions for the Ageing Populations – Addressing Global Challenges)*, Honolulu, Hawaii, 2020, https://arxiv.org/abs/2008.11834

- **Williams, A.S**., Ortega, F. (2020), "Insights on visual aid and study design for gesture interaction in limited sensor range Augmented Reality devices," In *IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 19-22, 2020, DOI: https://doi.org/10.1109/VRW50115.2020.00286

- Ortega, F., Kress, M., Tarre, K., **Williams, A**., Rishe, N., and Barreto, A. (2019), "Selection and Manipulation Whole-Body Gesture Elicitation Study in Virtual Reality," In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (NIDIT)*, Osaka, Japan, 2019, pp. 1110-1111. DOI: https://doi.org/10.1109/VR.2019.8798182 - Short paper

- Ortega, F., Kress, M., Tarre, K., **Williams, A**., Rishe, N., and Barreto, A. (2019), "Selection and Manipulation Whole-Body Gesture Elicitation Study In Virtual Reality," In *IEEE Conference on Virtual Reality and 3D User Interfaces (VR) (NIDIT)*, Osaka, Japan, 2019, pp. 1723-1728. DOI: https://doi.org/10.1109/VR.2019.8798105 - Workshop Paper

Books

- **Williams, A. S.**, & Ortega, F. R. (2021). A Concise Guide to Elicitation Methodology. arXiv e-prints, arXiv-2105. https://arxiv.org/abs/2105.12865

Magazine Articles

- **Williams, A.S.**, and Ortega, F.R. (2020) "Evolutionary gestures: When a gesture is not quite legacy biased". In *ACM interactions 28*, 4 (October - September 2020), DOI: https://doi.org/10.1145/3412499

Courses

- **Williams, A.**, and Ortega, F.. (2022). "An Introduction to Elicitation Study Design ". In *Human Computer Interaction International (HCII 2022)*. Human Computer Interaction International, Virtual, USA, **(Upcoming: 06/26/22 - 07/01/22)**

- Ortega, F., **Williams, A.**, and Garcia, J. (2020). "Multi-modal gesture elicitation methodology for children". In *Proceedings of the 2020 ACM Interaction Design and Children Conference: Extended Abstracts (IDC '20)*. Association for Computing Machinery, New York, NY, USA, pp. 85–88. DOI: https://doi.org/10.1145/3397617.3401808

Masters Degree Thesis

- **Williams, A.** (2020). "The Impact of Referent Display on Interaction Proposals During Multimodal Elicitation Studies". *Colorado State University*. https://hdl.handle.net/10217/232528, **(Embargo expiration date: 06/02/2022)**

A.2 Questions Asked During Phase 2

The questions below were asked to participants during phase 2 of the AR-VR comparison study (Chapter‘4).

- Which manufacturer or manufacturers produce more than half of their cereals with greater than average fiber?

- Which manufacturer or manufacturers make the cereal with the highest fat content?

- Which manufacturer or manufacturers have greater than, or equal to, the average number of carbs in all of their cereals?

- Which would be more likely to have above-average protein: a cereal with above-average fat, or one with above-average carbs?

- Which manufacturer or manufacturers produce more cereals with above-average fat than below-average fat?

- Which manufacturer is most likely to make the cereal with the highest calories?
- Which manufacturer or manufacturers have the largest portion of their cereals containing lower than average fat?
- For the manufacturer Kellogg's what is the correlation between sugars and carbs?
- What sugar content in grams has nearly the same number of points with above-average protein as below average protein?
- The three cereals with the highest fiber have above or below average carbs?
- Is more fiber more strongly associated with carbs or sugar?

A.3 Technical Details

The choice to make this system work across computing platforms greatly expands its utility but also increased its development effort. Cross-platform design means that the system has to be able to be built for different architectures (i.e., windows universal platform, windows/Linux standalone). The major benefit of cross-platform use is that the system can use input devices with technologies that were not intended to accept those inputs. In this system this is seen when using a Vive controller or the Logitech VR-Pen while wearing an AR-HMD, allowing the AR-HMD to utilize the benefits of the 6DoF inputs that VR devices provide.

That choice necessitated the use of a multiplayer networking system and a method of synchronizing the world spaces of the devices used. This was difficult because the Hololens 2 used a vision-based world mapping system that sets its coordinate space's origin at the location the application was started. VR devices use a world origin that is set up in a separate system, typically SteamVR.

The first implementation of the system used to synchronize and network across devices was published at a workshop during the 2021 IEEE Conference on Virtual Reality and 3D User Interfaces. This paper is included below, followed by a section outlining changes made to that system after the publication of that paper.

A.3.1 Paper: Cross-Device World and Input Synchronization

Title Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment

Authors Adam S. Williams and Francisco Ortega

Publication Venue This paper was originally published at the IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) [131].

Two of the key features of this research platform are that it can be run across different devices and that multiple people can collaborate in the environment. To put this into better context, a person can put on a VR-HMD and see the visualization. They can annotate it with their controllers and the provided tools. Later a second user can join the same session using an AR-HMD. If these two users are co-located each device needs to synchronize its world coordinate system with one another. This way all of the content viewed in one device can been seen in the same location by the other device. In addition to allowing co-located collaboration this also lets users share inputs to the system. The person with the VR-HMD can let the person in the AR-HMD use their tracked 6DoF input. A technical guide to setting up the first iteration of this synchronization process was published at the 2020 IEEE VR NIDIT conference [131]. This paper is provided below.

Title Using a 6 Degrees of Freedom Virtual Reality Input Device With An Augmented Reality Headset In A Collaborative Environment

Authors Adam S. Williams and Francisco Ortega

Abstract

Augmented reality headsets have become increasingly consumer-available. Often gesture and speech are the main input modalities provided by these headsets. For some tasks, users may need a more precise input method. Tracked controllers can be added by using image tracking; however, this is not always the most accurate solution. This work outlines how to use off-the-shelf products

to create a collaborative cross-device mixed reality experience. In that experience, the positionally tracked inputs from one headset can be used by another headset that may not natively support them.

Introduction

Augmented reality (AR) head-mounted displays (HMDs) are becoming increasingly popular, as products such as the Microsoft HoloLens 2 [15] (HL2) becoming more readily available. This trend is exhibited in the United States governments purchase of $100,000$ HL2 units for military use [109]. AR-HMDs are being used in an increasingly wide span of research, including work on situated analytics [132], multimodal input elicitation [30], and city planing [133]. Some consumer-available AR-HMDs (i.e., the Magic Leap 1 [16]) ship with a 6 degrees-of-freedom (6DoF) controller; however, others do not (i.e., HL2). The HL2 ships with mid-air gesture, speech, and gaze as its primary interaction methods. It's predecessor the HoloLens 1 (HL1) shipped with those same inputs with the addition of a single button clicker that was not tracked [17].

Access to the mid-air gestures used by these devices is often included in the software development kit (SDK) provided for each device. As an example, the Magic Leap 1 included 8 pre-defined gestures in the Lumin SDK. Even with these gestures being easily accessible to developers when used they require the user's hands to be tracked by the device's cameras. The tracking space provided by these cameras is often limited [134], which can make certain interactions difficult. Furthermore, gestures are not always the most precise interaction method for selection or manipulation tasks [135].

In some use-cases, the addition of a tracked mid-air controller would be beneficial to both user immersion and interactions. In AR adding a tracked input is commonly done by using image tracking targets that are attached to emulated pen [136]. The downside to this solution is that the pen can only be tracked while it is in range of the headset's front-facing cameras and while the image tracking targets are visible to that camera. This type of tracking may cause drift or errors

[15] www.microsoft.com/en-us/hololens

[16] www.magicleap.com/en-us

[17] www.docs.microsoft.com/en-us/hololens/hololens1-clicker

where the image tracked input is not well calibrated to the actual location of the physical input. A second, more problematic issue is that this input can only be used while it is held in front of or within the tracking area of the AR-HMD. This can be limiting to users, especially so when the added input is a pen, where users may wish to write notes on a surface in front of them while looking at another location. Consider taking notes in a lecture while looking at the whiteboard.

This limitation may not be detrimental for all use cases. For web-browsing, streaming videos, and simple games, the combination of gesture and speech may provide plenty of interaction technique options. Other use cases, such as the selection and manipulation of nodes on a scatter plot in an immersive analytics environment, may require a more robust tracked input solution. This necessity is even more prevalent in applications that require a user's gaze to be in one location while they hold, or manipulate an object at a different location. As an example, if an analyst is writing notes while viewing a complex data structure in an AR-HMD they may wish to be able to write on a surface in front of them while glancing back and forth from that surface to the data representation they are analyzing. In this case, a camera tracked input would lose tracking when the user looks away from their hand (e.g., the image tracked input).

The main contributions of this solution are:

- A networked cross-device environment for standalone use, co-located collaboration, and remote collaboration
- The ability to use VR controllers with the HL2 or other MR devices
- Easy integration of less standard VR inputs such as the Logitech VR-Pen [18]

System Design

There are various commercially available AR-HMDs on the market, with a reasonable market share held by the Microsoft HL2. The HL2 is developed to run using the Universal Windows Platform (UWP) and the Mixed Reality Toolkit (MRTK) which makes integrating Steam virtual

[18]https://www.logitech.com/en-us/promo/vr-ink.html

reality devices difficult due to their use of a different MR toolkit (i.e., Steam VR) and different tracking systems (e.g., base-stations compared to on device tracking).

To remedy these issues we present an input solution using the Unity 3-dimensional (3D) development engine. This solution allows the use of any 6DoF VR base-station tracked inputs to be used with the HL2 or other MR-HMDs.

At a high level, this solution uses a multi-user client-server architecture to network various devices into the same virtual experience. The positions of these devices are tracked and centered relative to a fixed physical location. This allows synchronous co-located collaborators to view the same virtual environment in real-time. All connected HMDs can view and interact with the same virtual content; however, the world synchronization step differs by device. The instance running on the VR-HMD with the desired inputs can be connected and left on, while the controllers for that VR-HMD are given to the AR-HMD user, which in this project was a HL2. This allows the HL2 user to utilize the 6DoF input as tracked by the VR-HMD's base stations (e.g., infrared tracking stations). The VR-HMD used in this project was the HTC Vive-Pro [19].

Synchronization is achieved on the AR-HMD by using Vuforia image tracking to align a virtual anchor with a real-world location. This anchor is referred to as the synchronization anchor, which is an empty game object. This anchor and its corresponding Vuforia image target are shown in Figure A.1. VR-HMDs place their synchronization anchor at this real-world location by using one of the trackers provided by the device (the black object in Figure A.1). This project used one of the base-stations provided by an HTC Vive-Pro. GameObjects can be parented in that synchronization anchor and have their locations relative to that anchor synchronized between devices.

Networking the devices adds a need to share the object locations over the network, which can add some latency. The benefit of this approach is that the networked solution provides an expandable synchronized collaborative environment, where more than one user may log in from different MR devices.

[19]https://www.vive.com/eu/product/vive-pro/

Figure A.1: Vuforia image target mounted on top of Vive-Pro Base Station 2.0, synchronization anchor position adjusted down from the center of the image target to the base station center.

Related Work

When looking at input devices for object selection and manipulation in MR, work has found that mid-air pen outperforms standard vive controllers in terms of speed and user preference [137]. There is also evidence that pen-like input devices can outperform finger-pointing (i.e., gesture) inputs [135], and that controller based hand-tracking has outperformed the mouse in some 3D positioning tasks [138]. That controller and pen based inputs show promise in MR environments motivates the use of a mid-air pen or other tracked controller with an AR headset.

In 2017 Bai et al. outlined an approach for using a Vive VR-HMD 6DoF controller with an HL1 AR-HMD [139]. That solution used image tracking to synchronize the HL1 with a fixed world position. The Vive HMD was also similarly synchronized by using one of the Vive controllers placed at the same location as the HL1 tracked image in the real-world. Bai et al. then used an off the shelf solution for Bluetooth networking to transmit the Vive controller coordinates and information to the HL1 [139]. This work differs in a few major ways. First, the devices and software used are different simply due to the time that has passed since the publication of that paper. That difference also leads to the necessity of the shared anchor. The older solution was able to move the world origin to synchronize virtual content. The most important difference is that the previous solution is for a single device where this solution can allow several devices to connect to the same environment. The basic implementation of this solution allows 20 clients. Some minor modifications can be made to the server (i.e., local hosting) to allow up to 100 clients to join the same experience.

Outside of using tracked controllers, other work has used image tracking along with a tangible object, such as a 3D printed pen [136]. While those image tracked solutions are viable, they can lack precision and the ability to be used when the images are not in view. Additionally, most tracked controllers can provide haptic feedback which may be desirable for some projects. Another solution for mapping a 3D model in a VR environment to a real-world environment is to find three or more real-world locations and to use the VR-HMD's controllers to locate and record those points, after which the system can adjust the alignment of those same three points in the virtual

model to those points in the real-world [140] [20]. A similar approach could be used here where more than one image target is used to synchronize the devices which would help to reduce the calibration step used here.

System Components and Integration

This section outlines how to integrate the various components needed for this solution first. Later more detail is provided on the setup steps for each component. This solution uses the following off the shelf components: Unity (version 2019.3.1f LST), Photon Networking version 2 (Photon PUN 2), Vuforia image tracking, the MRTK (version 2.5), a Microsoft HoloLens 2, an HTC Vive-Pro with base stations and controllers, and a Logitech VR-Pen.

Unity The first step to setting up this solution is to install the Unity game engine. Unity is a game development engine that supports development for most MR-HMDs. This work uses the Unity version 2019.3[21]). Unity may be downloaded at https://unity3d.com/get-unity/download.

Mixed Reality Toolkit This project uses the MRTK version 2.5 [22]. The MRTK is a multi-platform mixed reality SDK that is compatible with the HL2, Windows mixed reality, open VR, and most consumer-available MR devices. The recommended way to integrate the MRTK into the Unity project is to add the required packages to the project manifest file. This process is further detailed in the MRTK documentation[23].

Photon Engine Next, the networking software needs to be added. This project uses Photon engine 2 [24]. Photon engine is a Unity compatible multiplayer networking system that offers both free and paid usage options. The free solution offers access for up to 20 networked devices, or 100

[20]https://github.com/felixkosmalla/unity-vive-reality-mapper

[21]https://unity.com/releases/2019-3

[22]https://microsoft.github.io/MixedRealityToolkit-Unity/

[23]https://microsoft.github.io/MixedRealityToolkit-Unity/version/releases/2.5.0/Documentation/usingupm.html

[24]https://www.photonengine.com/

networked devices if hosted on your own server. To add Photon to the Unity project go to the asset store and then search for and import "PUN 2 - Free" [25].

Image Tracking This step is not necessary unless the project uses devices without provided trackers (i.e., HL2). A simple to use image tracking solution is provided by Vuforia engine [26]. Vuforia is a free to use image tracking system that offers several types of image tracking options. This project will only use the basic image tracking capabilities. In Unity 2019 Vuforia can be added to a project in the "Project Settings →Player" section. To add it check the "Vuforia Augmented Reality Supported" check box. Note that Vuforia is not supported in "windows standalone builds", which are often used when deploying to VR-HMDs. If a VR-HMD is used, the location of a provided device tracker can be used in-place of Vufoira. This project uses the location of one of the provided Vive base-stations.

System Setup

This section provides steps for how to use the above-outlined components in unison towards the goal of creating a synchronous collaborative cross-device experience.

Networking Setup

First, set up the project to start a networked game instance. Once an instance is initiated add the connected clients as players by using Photon engine's "instantiate" function. This will require setting up a networked lobby, room, and player. An overview of how these can be set up can be found in the Photon Unity tutorial [27]. Other details more specific to using Photon with the MRTK can be found in the multi-user MRTK tutorial[28]. When tested on a residential network the latency encountered when synchronizing objects was around 78.6 milliseconds (ms) when averaged over

[25]https://assetstore.unity.com/packages/tools/network/pun-2-free-119922

[26]https://developer.vuforia.com/

[27]https://doc.photonengine.com/en-us/pun/v2/demos-and-tutorials/pun-basics-tutorial

[28]https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials

1000 updates. Prior work using 2D selection tasks found that this level of latency can cause minor (e.g., 15% performance decrease at 40 ms, 50% performance decrease at 225 ms) performance losses; however, those results were not directly extended into a 3D environment [141].

Coordinate Synchronization

Often the world origin (e.g., world-space coordinate 0,0,0) for AR-HMDs the origin is set based on where the HMD is turned on or where the HMD was when the app was started. VR-HMDs may have a world origin set in the same way or set based on a location set up in the "set up playspace" step of configuring the headset. Changing this location is not an optimal solution for synchronization and is discouraged by the MRTK [142, Section 4]. A better solution is to use a Unity GameObject with its location set to the desired world anchor (i.e., the synchronization anchor). Any networked objects that need to be synchronized can be parented in that anchor GameObject. While this type of object synchronization is not provided by Photon by default, it can be added with relatively low effort. When synchronized objects are the local instance they will need to send their transform information to their networked corollaries. When these networked instances receive that information they can be set to update their position accordingly.

The steps for placing a shared anchor in the appropriate real-world location differs based on the type of devices being used. These differences fall into two major categories: devices with image tracking and devices with position trackers. Some part of the decision to use one method over the other may be influenced by the build target used. When using the HL2 or other Windows MR devices the build should be set to UWP. When using Steam VR based or other VR based devices the build should be set to "PC, Mac, & Linux Standalone".

The two different build targets have access to a different set of functionalities in the Unity engine application programming interfaces (API). These differences can be seen in the Unity documentation under "unityEngine.XR". For the scope of this paper, the most important difference in functionality is that the UWP does not have access to base-station locations. We recommend using

Unity's "Platform Dependent Compilation symbols" [29] if the project will be run from both build targets. These allow the specification of which parts of code get compiled for which build targets.

Synchronization of Devices With Trackers Unity XR nodes and device trackers can be used to set the location of the real-world aligned synchronization anchor. This is most easily done by accessing the information of the tracked device through its unity XR node and centering the synchronization anchor on it. For this solution, one of the two base-stations used by the HTC Vive-Pro is used; however, the use of a controller or other tracker can be similarly effective [139]. Within the scope of this project, no interference between the HL2 tracking and Vive Pro 2.0 base stations was noticed. More information on how to locate XR tracked objects in Unity is listed under "XRNodeStates" in the Unity documentation[30].

Image Tracking Based Synchronization Vuforia should be used with devices that have image tracking capabilities but no positional trackers. Vuforia requires that an image target is set up before tracking is started. This process is outlined in their tutorial'[31]. Once the image is registered for tracking with Vuforia it can be printed and affixed to the desired position of the real-world location for the synchronization anchor. In this project, that location was on-top of the Vive base station. This setup is shown in Figure A.1.

This project only uses Vuforia once, to locate the base station image target. Once the target has been found, Vuforia is no longer needed. To conserve computational overhead we recommend manually enabling and disabling Vuforia so that it is only active once. Allowing manual enabling on Vuforia can be done by setting Vuforia to delayed initialization. The option for that is option is located in the Vuforia settings. For this project, we used platform dependent compilation symbols to enable Vuforia when an AR-HMD was connected and the appropriate scene (i.e., level) was loaded. Vuforia was then disabled upon manual acceptance of the identified image target. If a lot

[29] https://docs.unity3d.com/Manual/PlatformDependentCompilation.html

[30] https://docs.unity3d.com/ScriptReference/XR.InputTracking.html

[31] https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity.html

of headset movement is expected, it may be best to leave Vuforia tracking on or to set up an MRTK spatial-anchor to continually adjust the location of the anchor in case of tracking drift.

The accuracy of the tracked controller inputs depends on the accuracy of the image target and its tracking. We have found that a full page size image provides a reasonably easy to find target. The placed anchor can drift when the headset is used too far from the placed anchor or when the task involves looking around. This is caused by minor changes in the location of the spatial mapping that the HoloLens uses to track the location of the target. When a lot of headset movement is expected we recommend keeping Vuforia on to allow re-centering the synchronization anchor. An alternative approach would be to use the azure spatial anchors that are provided by the MRTK to maintain the synchronization anchor's position.

Cross-Device World Synchronization When both image targets and the device's positional trackers are used for world alignment, some minor positional adjustments may need to be made. For this project, the position of the image target centered synchronization anchor has been adjusted down a fixed amount to accommodate for the difference in height between the base-station tracked location, and the image target affixed to the top of the base station. Visualizations of the controllers and the connected HMDs are shown in the experience to make the adjustment process easier between devices. These models are shown as faint blue outlines with transparent gray bodies for the controllers and as a 3-axis GameObject with a user or device name displayed above it for connected headsets. These models are shown in Figure A.2.

Tracking Adjustments Photon engine provides object synchronization natively. However, due to the implementation of the world aligned anchor, the Photon provided synchronization will not work. Custom photon information streams can be set up to achieve cross-device synchronization. This process is briefly described in the MRTK multi-user tutorial [32]. This tracking will either send the object's transform information relative to the synchronization anchor (e.g., the objects local transform), or it will receive that information and adjust its own transform.

[32]https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials/

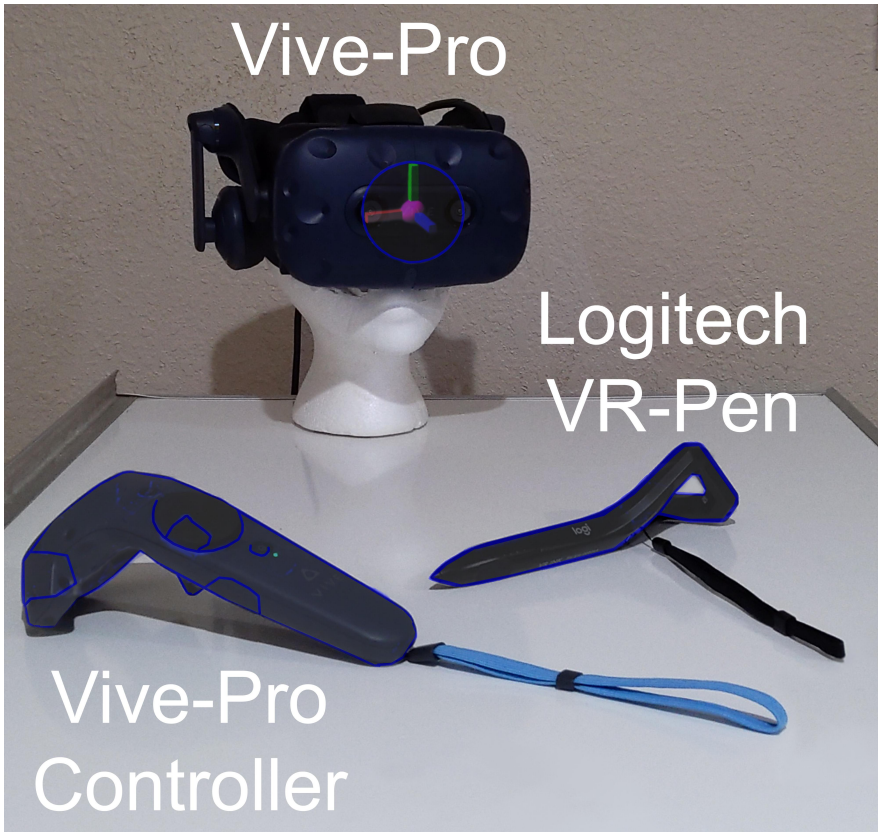


Figure A.2: Vive-Pro client and controllers as viewed through the HoloLens 2.
Legend: The faint blue / gray controller outlines are the rendered controller representations to visualize cross-device synchronization, the controller labels are not part of the system, the Vive-Pro label is enlarged for legibility.

Note that as of the publication of this guide (2021), some users may have issues where the devices will connect to Photon but are unable to join the same instance of the networked environment. The solution is currently to uninstall the Windows SDK version 10.0.19041.0 and to use version 10.0.18362.0.

Controller Usage At this point, the project can synchronize content across devices. To use the 6DoF tracked controller with an MR-HMD that does not natively support it, both headsets need to be connected to the same instance of the running Unity application. When the headset that owns the desired controller is connected it will need to register its controllers with the application and then parent their virtual representations in the synchronization anchor. The controllers can then be used by the person wearing the other HMD. The inputs and positions from the controllers are still tracked by their HMD; however, as everything is synchronized they can now be accurately used by any party present in the same physical space.

Mid-Air Pen Use-Case

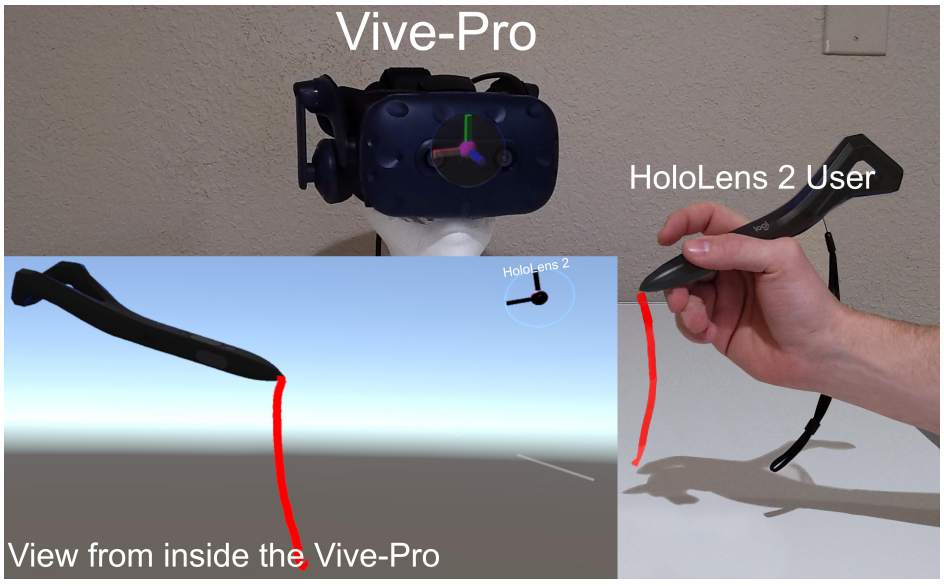


Figure A.3: A HoloLens 2 user drawing a line using the VR-Pen as seen by both the HoloLens 2 user and the Vive-Pro user

Legend: Bottom left: view from inside the Vive-Pro with the HoloLens user shown in the upper right, outside of the Vive-Pro view: the HoloLens 2 user's view

To demonstrate the cross-device functionality of this solution we have implemented the Logitech VR-Pen[33] for basic line drawing on the HL2. An example of this functionality is shown from both the Vive-Pro user's and the HL2 user's viewpoint in Figure A.3. This was done by setting up the project according to the above-mentioned specifications.

After that setup Logitech VR-Pen can be accurately tracked with its position being nearly identical in the Vive and the HL2. To ensure that this tracking is functional while using an AR-HMD, a virtual model of the pen is rendered over the physical pen on each client (Figure A.2). When the tracking is not aligned properly this model will appear out of place compared to the physical pen (Figure A.4).

To enable the pen to draw, and to have that drawing seen in real-time on both devices, Unity's "Line Renderer" class is used along with a separate class for handling synchronization. The line renderer class creates a line that connects a provided set of coordinate locations. To render the line on each display as points are recorded those points must also be sent and added to each networked instance of the line render component. This can be achieved by sending the points as part of the data stream that includes the local position of the pen. In this project, all of the point and location transmissions are handled by a separate synchronization class attached to the line render component. The line drawn can be seen from both the Vive and HL2 user's viewpoint in Figure A.3 and from the HL2 user's when not properly synchronized in Figure A.4.

For this project, the Vive-Pro headset remains usable with a single 6DoF controller. The HL2 user can use the VR-Pen, which is the Vive HMD's second controller.

Discussion

This solution outlines how to integrate and modify several free resources to create a collaborative MR experience and to allow the use of controllers across devices. The steps used to align the devices' world-locations is only relevant when setting up co-located collaborative environments. When creating remote collaborative environments the synchronization provided by Photon will be

[33]https://www.logitech.com/en-us/promo/vr-ink.html

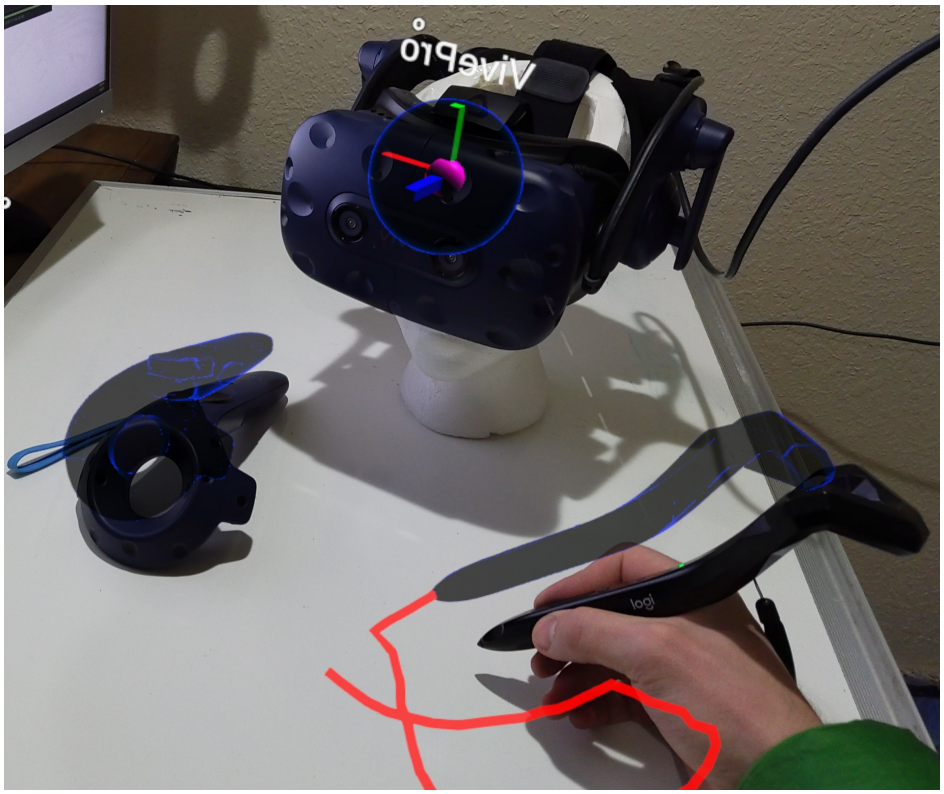


Figure A.4: Poorly synchronized Vive-Pro client and controllers as viewed through the HoloLens 2.
Legend: The faint blue / gray controller outlines are the rendered controller representations to visualize cross-device synchronization, the red is the line drawn using the VR-Pen.

enough. That said, the use of the custom information streams may still be beneficial as seen in the case of adding line renderer points when drawing a line in real-time. This solution is robust enough to handle a wide set of needs by accommodating both image tracker and positional tracker synchronizations as well as a selection of MR devices.

It is reasonable to assume that as these devices mature, the tracked area provided by them will improve. While the exact specifications of most AR-HMDs tracking and viewing areas are difficult to find, Microsoft has mentioned that the HL1 had a field of view (FoV) of 34-degrees which increased to 52-degrees in the HL2 [143]. Note that the FoV is different from the area tracked by the device. Even so, a similar trend of improving the tracked area in each device iteration can be expected. Yet, as of now, these devices offer limited tracking, and other AR-HMD product releases in the near future may include even less tracking as necessitated by a smaller form factor (i.e., Apple Glasses). This tracking limitation is also present on current mobile AR solutions.

Conclusion and Future Work

This paper presents a networked synchronous cross-device MR solution that allows users of MR-HMDs to use the controllers provided by other MR-HMDs located in the same physical space. This opens up opportunities for researchers to use 6DoF inputs with devices that typically do not support their use. A separate benefit of using this solution is that the devices connected are instanced into a shared synchronous MR experience, allowing this system to be used for research on collaborative environments.

This solution is a work in progress and can be improved in several ways. All of the networking was done over the cloud through Photon Engine. That choice enabled remote collaboration and helped to enable cross-device support; however, it was at the cost of increased latency. Setting up a private server or if the intended use-case allows setting up a local area network could both decrease this latency. This project also used Vuforia for image tracking. It is possible to incorporate a custom image tracking implementation if more control over it is necessary. Another path forward would be to set up the image tracked synchronization anchor alignment to work on a VR-HMD's

built-in cameras. This would remove the need to adjust the image target to the positional tracker's location.

While there is room for improvement, this solution uses an easy to assemble collection of off the shelf hardware's and software's to facilitate synchronous cross-device collaboration and controller use. This project can be easily extended for use in immersive analytics, collaborative AR, and other multi-device MR experiences. This ease of use is also found for extending the project to work with a variety of controllers as demonstrated by the use of the Logitech VR-Pen by an HL2 user.

A.3.2 Improvements to Coordinate Synchronization

The first design used a single image target mounted to the Vive base station. Using this image target the Hololens could center its synchronization anchor with the VR-HMD's base station. The Vive, having access to the base station location, could do the same. This design required a user to manually fine-tune the position and rotation of the recognized image target. To correct this misalignment the researcher had to don the AR-HMD and monitor the virtual controller's models against the physical controller while adjusting the rotation and position of the image target aligned synchronization anchor.

Objects in the AR-HMD are susceptible to drift, which is when an object slowly moves relative to the real world. Drift is most likely to happen when the AR-HMD is not looking at the object. The drift in itself was not a huge issue, often being a fraction of an inch. The larger issue was rotational drift. With the image target fixed to the base station and the base station above and behind the AR-HMD, the recognized image target was likely to drift. Minor variations in its rotation were magnified by the distance from the synchronization anchor to the controllers being used. With enough drift, the system would need to be re-calibrated.

To solve this, three full-page image targets were mounted to a 23-inch by 33-inch black mouse pad. When using the AR-HMD the image targets would be viewed and recognized by Vuforia. On recognition, the synchronization anchor was placed in the center of the three targets with a forward

vector facing the furthest image target. In the VR-HMD, the user now had to take their controller, place it over the image targets to match an outline printed in the center of them, and press the trigger button to record the location. The rotation and position of the synchronization anchor were then set using the same process as the AR-HMD.

This system provides several major advantages over the old system. Most importantly the researcher no longer needs to don the AR-HMD to perform the fine-tuning step. This adds a layer of COVID safety to the system, now only the participant needs to wear the AR-HMD. Additionally, the synchronization target is placed in the center of the participant's work-space, meaning the AR-HMD is less likely to look away from it. This positioning caused the rotational drift to be less likely and less magnified due to the shorter distance between the synchronization anchor and the controllers used.

A.3.3 Object Synchronization Improvements

Objects are synchronized by updating their transforms to maintain their relative position in relation to the synchronization anchor. The final version of this script uses linear interpolation to decrease network traffic by sending fewer updates and smoothing object motion between them. At each tick, the script checks to see if the current transform is greater than a user-provided minimum distance from their last transform. Rotations were checked by comparing the degrees of change. If those conditions were met, the new position is recorded as the last position and an update event is sent to all instances of the object. The objects receiving the updated transform start a co-routine that uses linear interpolation to slowly transition from their current position to the received position. If the local object is still being moved it continues to check the distance between the last transform and the current one and sends the new transform when appropriate. Upon receiving a new transform remote objects will interrupt their current co-routine and start a new one using their current location and the received transform.

This design decreased the number of updates required to be sent from one frame to one every half-inch (or other provided distance) of movement. The script uses the object's local (relative)

location when moving and performing distance checks. This means that depending on parent-child relationships implemented, fewer updates can be sent. As an example, consider a visualization that currently has ten annotations placed on it. If those annotations are not parented under the visualization, any update to the visualization would require eleven transform events, one for each annotation, and one for the visualization. By parenting the annotations in the visualization we can send one update event for the visualization because the annotation's positions relative to the visualization are unchanged.

An additional benefit of the custom event-driven synchronization system is that it was object ownership agnostic. When using Photon engine objects can be "instantiated" where they will be loaded on all clients but are owned by the client that called for them to be instantiated. Objects can also be "room instantiated" where any client can call for them to be loaded but the ownership of the object is assigned to the master client. Using a pun observable system, the updates are only sent by the owner where other clients need to request ownership transfers to interact with the objects. This event system allowed any client to interact with any object.

A.4 Surveys

A.4.1 Short Graph Literacy Scale Plus

The original short graph literacy scale and its images are provided by the open science framework [34] [122]. The additional scatter-plot specific questions asked during this work are shown in figures A.5A.6A.7.

The additional scatter-plot questions used are shown in Figures.

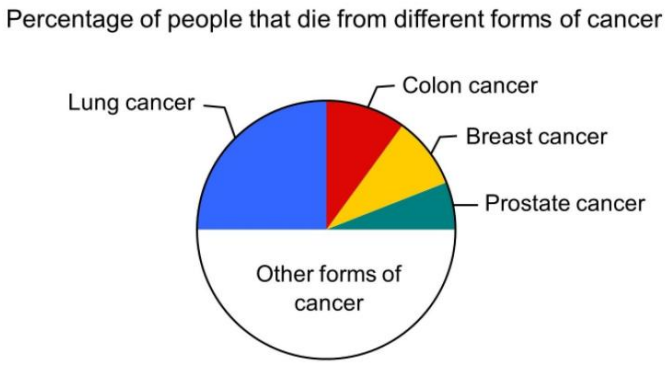
A.4.2 Paper Folding Test

The paper folding test originally was originally released in the "Kit of Factor-Referenced Cognitive Tests" in 1976 [2]. The version used in this study is shown in figures A.8A.9A.10.

[34]https://osf.io/frjbq/

Item 1

Here is some information about different forms of cancer:



Approximately what percentage of people who die from cancer die from colon cancer, breast cancer, and prostate cancer taken together?

Your answer

Back Next Clear form

Figure A.5: Scatter-plot graph literacy question 1

Item 2

In a magazine you see two advertisement, one on page 5 and another on page 12. Each is for a different drug for treating heart disease, and each includes a graph showing the effectiveness of the drug compared to a placebo (sugar pill).

Compared to the placebo. which treatment leads to a larger decrease in the percentage of patients who die?

- ○ Crosicol
- ○ Hertinol
- ○ They are equal
- ○ Can't say

Back Next Clear form

Figure A.6: Scatter-plot graph literacy question 2

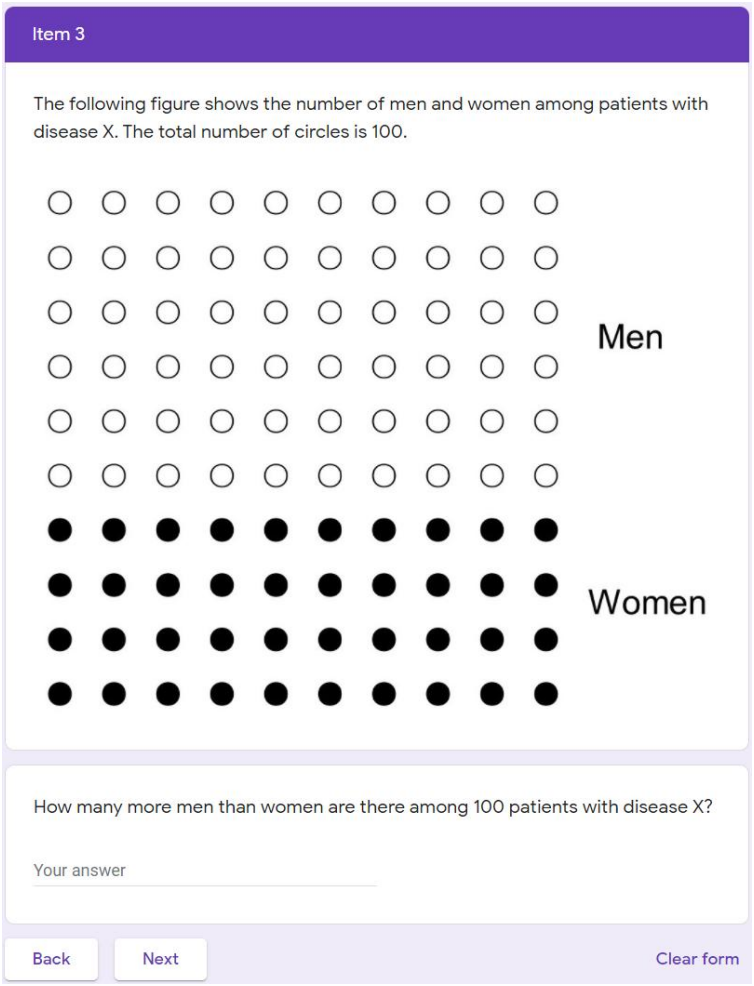
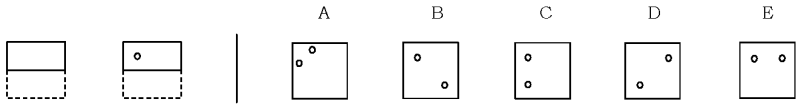


Figure A.7: Scatter-plot graph literacy question 3

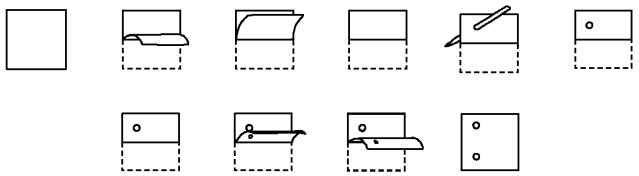
Paper Folding Test—Vz-2-BRACE

In this test you are to imagine the folding and unfolding of pieces of paper. In each problem in the test there are some figures drawn at the left of a vertical line and there are others drawn at the right of the line. The figures at the left represent a square piece of paper being folded, and the last of these figures has one or two small circles drawn on it to show where the paper has been punched. Each hole is punched through all the thicknesses of paper at that point. One of the five figures on the right of the vertical line shows where the holes will be when the paper is completely unfolded. You are to decide which one of these figures is correct and draw an X through that figure.

Now try the sample problem below. (In this problem only one hole was punched in the folded paper).



The correct answer to the sample problem above is C and so it should have been marked with an X. The figures below show how the paper was folded and why C is the correct answer.



In these problems all of the folds that are made are shown in the figures at the left of the line, and the paper is not turned or moved in any way except to make the folds shown in the figures. Remember, the answer is the figure that shows the positions of the holes when the paper is completely unfolded.

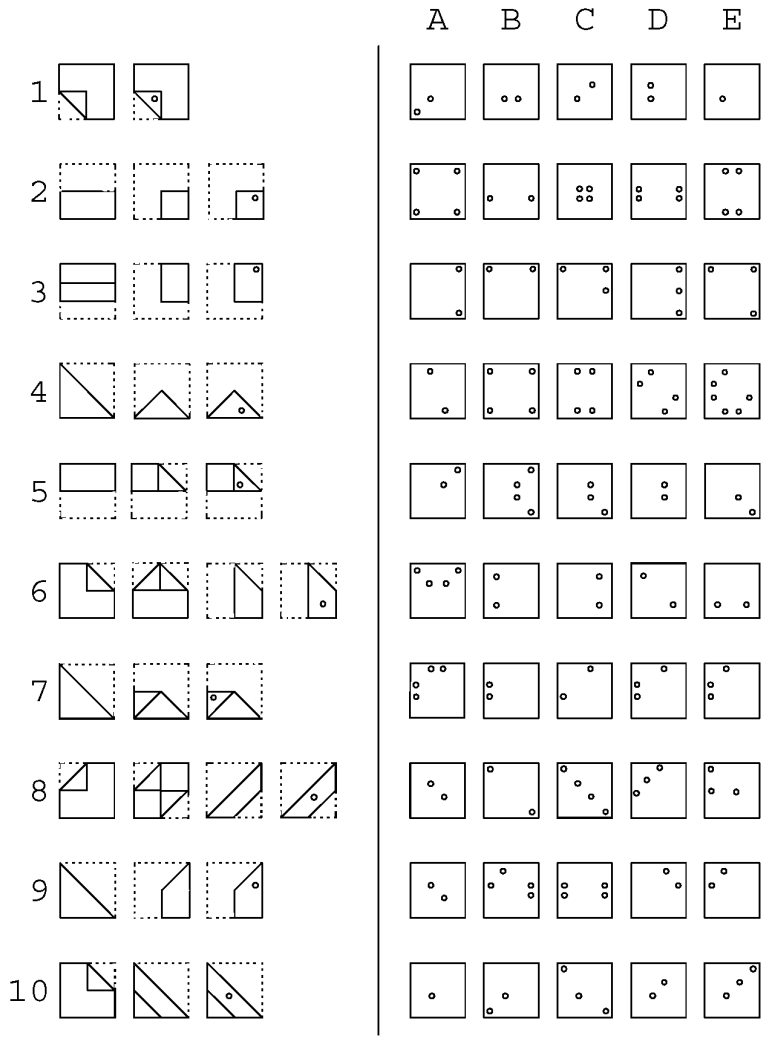
Some of the problems on this sheet are more difficult than others. If you are unable to do one of the problems, simply skip over it and go on to the next one.

You will have three minutes for each of the two parts of this test. Each part has one page. When you have finished Part One, STOP. Please do not go on to Part Two until you are asked to do so.

DO NOT TURN THIS PAGE UNTIL ASKED TO DO SO

Figure A.8: Paper Folding Test VZ-2, Page one [2]

PART ONE (3 MINUTES)



STOP

DO NOT PROCEED TO THE NEXT PAGE UNTIL ASKED TO DO SO

Figure A.9: Paper Folding Test VZ-2, Page two [2]

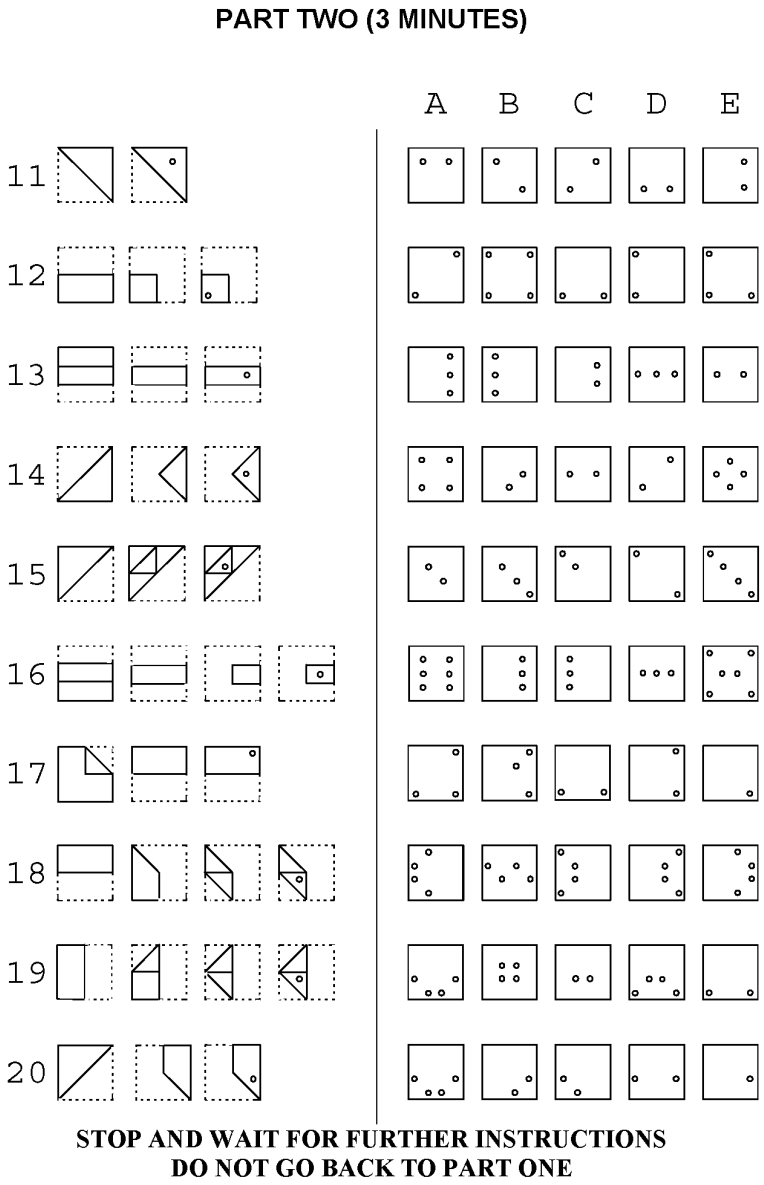


Figure A.10: Paper Folding Test VZ-2, Page three [2]

A.5 Data-set

The Cereals dataset was taken and modified from the version provided with the IATK [1]. The modified dataset is shown in tables A.1 and A.2.

Table A.1: Cereals dataset part 1, condensed from the version provided with the IATK source code [1]

Manufacturer	Fat	Dietary Fiber	Carbs	Sugars	Protein g
Kelloggs	2	3	17	13	3
Kelloggs	1	2	20	9	3
Kelloggs	2	3	21	7	3
General Mills	1	4	15	14	3
General Mills	2	2	18	8	3
General Mills	2	1.5	13.5	10	3
Quaker Oats	5	2	8	8	3
Quaker Oats	2	0	12	12	1
General Mills	3	0	13	9	1
Kelloggs	0	5	14	12	3
Post	3	3	13	4	3
Quaker Oats	2	1	12	11	1
Kelloggs	1	0	15	9	2
Post	1	6	11	14	3
Kelloggs	1	5	14	12	3
Ralston Purina	2	1	14	8	2
General Mills	2	1.5	10.5	10	2
Kelloggs	0	1	11	14	2
General Mills	2	2	17	1	6
General Mills	2	2	13	7	3
General Mills	1	0	12	13	1
Ralston Purina	0	0	22	3	2
Kelloggs	0	1	13	12	1
General Mills	1	0	12	13	1
Kelloggs	3	4	10	7	3
Kelloggs	0	1	21	3	2
Kelloggs	1	1	11	13	2
Kelloggs	0	1	14	11	1
Post	1	0	13	12	1
General Mills	1	0	15	9	1
Post	0	3	17	3	3
General Mills	1	1.5	11.5	10	3
Post	0	0	14	11	1
Kelloggs	1	1	17	6	2
General Mills	1	0	21	3	2
General Mills	1	0	12	12	2
Ralston Purina	0	0	23	2	1
Kelloggs	0	0	22	3	2
Kelloggs	1	1	9	15	2
Kelloggs	0	1	16	3	6

Table A.2: Cereals dataset part 2, condensed from the version provided with the IATK source code [1]

Manufacturer	Fat	Dietary Fiber	Carbs	Sugars	Protein g
General Mills	1	0	21	3	2
General Mills	1	0	21	3	2
General Mills	1	0	13	12	1
General Mills	1	1	16	8	2
Kelloggs	0	1	21	2	2
Nabisco	0	1	21	0	3
General Mills	1	2	11	10	2
Ralston Purina	0	1	18	5	2
Kelloggs	0	3	14	7	3
Post	0	0	11	15	2
Post	1	3	15	5	3
Quaker Oats	2	2	12	6	4
American Home Food Products	1	0	16	3	4
General Mills	1	2	15	6	2
Kelloggs	0	1	20	3	3
Quaker Oats	1	2	14	6	4
Quaker Oats	2	2.7	-1	-1	5
General Mills	2	2.5	10.5	8	3
General Mills	1	3	16	3	3
Ralston Purina	1	3	17	3	3
General Mills	1	3	17	3	3
Ralston Purina	1	4	15	6	2
Post	0	5	13	5	3
Kelloggs	0	3	18	2	3
Kelloggs	0	2	15	6	2
Nabisco	0	4	19	0	3
Nabisco	0	3	20	0	3
Nabisco	0	3	15	5	2
Nabisco	0	3	16	0	2
Nabisco	1	10	5	6	4
Kelloggs	1	9	7	5	4
Kelloggs	0	14	8	0	4
Quaker Oats	0	0	13	0	1
Quaker Oats	0	1	10	0	2