

UNIVERSITY OF CALGARY

Immersive Analytics Interaction: User Preferences and Agreements by Task Type

by

Qing Chen

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

MAY, 2018

© Qing Chen 2018

Abstract

For immersive computing environments, multiple interaction modes (e.g. voice, gestures, handheld controller) have been proposed. In this thesis, I present the results of an elicitation study examining user preferences and measuring interaction agreements, based on two task types from an existing task taxonomy, in the context of data interaction in augmented reality (AR). The results indicate a non-statistically-significant association between a user's input mode preference and the type of the performed task in most cases. However, agreements on interactions were found to be higher in one type of task. I reflect on the resulting implications and offer one practical guideline for UX designers creating AR-based analytics applications. This thesis also details an alternative way of quantifying user agreements in an elicitation study on interactions.

Keywords: speech input; gesture input; elicitation; immersive analytics; augmented reality; Microsoft HoloLens.

Acknowledgements

I want to thank my supervisor, Dr Frank Maurer, for first accepting me into his lab where I met a few highly intelligent but very friendly lab mates who were doing ambitious software engineering projects with cutting edge AR/VR technology. I also want to thank him for teaching me the software entrepreneurship course because one of the most important lesson I learnt from that course was how to prioritize work based on its value (not complexity!), without which I would not have been able to finish this thesis on time. Speaking of the thesis, Dr Frank helped me tremendously by first suggesting its topic, and then by offering me pragmatic, actionable guidance at the right time. I want to thank him especially for instilling in me statistical rigor as well as encouraging me to come up with an alternative formula to calculate user input agreement rates, which I consider to be a highlight in my thesis. Lastly, I want to thank him for the generosity of inviting both me and my family to his house, twice, for dinner around the Thanksgiving holiday. The house was really nice and inspired me to study even harder!

Dedication

This dedication is obviously for my wife, Ying, and my nearly 3-year-old boy, Helios. Graduate study was hard and time-consuming. There were inevitably some late nights which made you unable to see me for more than 24 hours. And I felt awful about that. Especially to my son: daddy felt terribly sorry every time I had to leave you behind to mommy when you slept, or ran away right in front of you – because I had to finish my study. But I am also happy to have both of you in my life because together, we are discovering and enjoying the true purpose of our lives.

Table of Contents

Abstract.....	ii
Acknowledgements	iii
Dedication	iv
Table of Contents	v
List of Tables	vii
List of Figures and Illustrations	viii
Chapter 1 Introduction.....	1
1.1 Three Distinct Input Modes	2
1.2 Specification and Manipulation Tasks.....	3
1.3 Research Goals	4
1.4 Research Questions.....	5
1.5 Research Methodology	5
1.6 Contributions	6
1.7 Thesis Structure	7
Chapter 2 Background and Related Work.....	8
2.1 Brief Overview of Input Technologies for Immersive Computing.....	8
2.2 Input Mode Preference.....	13
2.3 Elicitation Studies	16
2.4 Input Agreement	17
2.4.1 Quantification methods.....	17
2.4.2 Empirical studies.	20
2.5 Task Taxonomies	24
2.6 Where My Study Differs	28
Chapter 3 Elicitation Study Methodology	29
3.1 Terminology: Referent vs. Task	29
3.2 Task Selection.....	30
3.3 Apparatus	34
3.4 Procedure.....	35
3.5 Participants.....	38
Chapter 4 Elicitation Results	39
4.1 Data Size	39
4.2 Input Mode Preference.....	39
4.2.1 Statistical significance.	41
4.2.2 An aggregated view of reasons.	42
4.2.3 A sample view of individual reasons.....	43
4.3 Sitting vs. Standing While Gesturing.....	44
4.4 Input Agreement Rates	44
4.4.1 Definition of same inputs.	45
4.4.2 Clustering of same inputs.	47

4.4.3 An alternative formula for agreement rates.	47
4.4.4 Result.	51
Chapter 5 Discussion on Results.....	54
5.1 Answers to the Research Questions	54
5.1.1 Preference question.	54
5.1.2 Agreement question.....	56
5.2 Possible Explanation on Difference in Agreements	58
5.3 Implications from Differences in Agreements.....	58
5.4 Significance of Task Types.....	59
5.5 Comparing Agreement Rates Calculated with My Formula, Max-consensus and Vatavu <i>et al.</i> 's Formula	60
Chapter 6 Limitations, Future Work and Conclusion	63
6.1 Limitations.....	63
6.2 Future work.....	65
6.3 Conclusion	65
References	67
Appendix A: User Input Preferences	72
Appendix B: User Input Encodings	76
Voice Encodings	76
Gesture Encodings	81
Controller Encodings.....	86

List of Tables

Table 1 - Specification tasks used in the elicitation.....	31
Table 2 - Manipulation tasks used in the elicitation	32
Table 3 - Preference count for each input mode in all the 13 tasks	40
Table 4 - p-value concerning the difference of preference counts for all tasks	42
Table 5 - Characteristics of the three agreement rate metrics used in prior research	49
Table 6 - Input agreement rates for each input mode in all the 13 tasks	51
Table 7 - Breakdown of voice, gesture and controller agreement rates by task types.....	53
Table 8 - Most common inputs for each input mode in every task.....	53
Table 9 - p-values from McNemar's test relating only to the preferred input mode for each specification task.....	55
Table 10 - p-values from McNemar's test relating only to the preferred input mode for each manipulation task	55
Table 11 – Agreement rates according to Morris’ max-consensus.....	60
Table 12 - Agreement rates according to Vatavu <i>et al.</i> ’s formula.....	61

List of Figures and Illustrations

Figure 1 - Three distinct input modes: voice (left), gesture (middle), controller (right). Images from (left to right): https://www.voicebot.ai , http://www.gestigon.com , https://www.engadget.com	2
Figure 2 - left: an outfit with body sensors, data gloves and a head mount display; right: a data glove. Images from (left to right): http://theconversation.com , https://www.timetoast.com	9
Figure 3 - left: a user interacting with CAVE; right: a wireless wand used as the input device for CAVE. Images from (left to right): https://www.lifehack.org , https://cosmosmagazine.com	10
Figure 4 - left: a military parachuter in VR training environment, using parachute straps as the input device; right: a user in shooting training session with a model gun. Images from (left to right): http://www.vrs.org.uk , https://en.wikipedia.org	11
Figure 5 - Oculus Rift Touch controllers. Images from (left to right): https://www.oculus.com/	11
Figure 6 - the Leap Motion USB-like device. Left: mounted on a VR device; Right: used in a desktop environment. Images from (left to right): https://en.wikipedia.org , http://smartgimmick.com	12
Figure 7 - Microsoft HoloLens, with voice and gesture recognition built in. Right: a HoloLens-wearing user issuing gesture commands in front of mid-air hologram charts. Images from (left to right): https://www.microsoft.com/en-ca/hololens , https://sg.news.yahoo.com/	12
Figure 8 – Cabral <i>et al.</i> 's Gesture Usability Study. On the left: gesturing in CAVE; On the right: gesturing in front of a projector screen. Images from: [2]	13
Figure 9 – Mota <i>et al.</i> 's study comparing gamepad and gesture inputs. One the left: user with a gamepad; On the right: user interacts with an Oculus Rift where a LeapMotion sensor was attached. Images from: [6].....	14
Figure 10 – Pick <i>et al.</i> 's study where a participant was planning the layout of a factory floor in the CAVE immersive environment. Images from: [3].....	15
Figure 11 – Vatavu <i>et al.</i> 's freehand vs. handheld gestures study. Left: an illustration demonstrating how the gestures were acquired. Right: Wii Remote, the handheld device used in the study. Users were allowed to combine both the button-pressings and the motion gestures together. Images from: [20].....	20
Figure 12 – Piumsomboon <i>et al.</i> 's gesture study with an AR simulation. Left: a participant in the study; Right: the simulated AR experience from a participant's view, with the car	

visual provided by the HMD and the table view provided by the camera attached to the HMD. Images from: [12]	21
Figure 13 - The living room used in Morris' <i>Web on the Wall</i> study. Note there is a decoy Kinect placed on the TV. Images from: [17]	22
Figure 14 – Kühnel <i>et al.</i> 's smart home study. Left: the lab setup. Right: different gestures proposed by a participant in the study. Images from: [4]	23
Figure 15 - Screenshot of what a participant would see in HoloLens.	34
Figure 16 - Button layout of an Xbox controller. Image from: https://commons.wikimedia.org . 35	
Figure 17 – A study participant doing gesture for the task “single-select”.	36
Figure 18 – A snippet of preference data collected in the elicitations.....	40
Figure 19 - Another representation of the agreement data from Table 6. The specification tasks were explicitly indicated in brackets. Note the asymmetric nature of this graph due to the large amount of empty space left by those specification tasks, which visually summarizes their overall low agreement rates.....	52

Chapter 1 Introduction

Immersive computing, a catch-all phrase for virtual reality (VR) and augmented reality (AR), has been in commercial space since 1985, when a company named VPL Research developed a full-body suit integrated with dozens of sensors as well as a head mounted display [21]. However, it was not until almost 30 years later that the technology fully caught on with the public, when a virtual reality device named Oculus Rift [29] and an augmented reality based game Pokemon Go were introduced by Facebook and Nintendo, respectively. Ever since then, we have witnessed an enormous advancement in the VR/AR hardware technology, from low resolution display to high resolution, from limited angle tracking to full 360-degree tracking. But those incremental improvements were mostly made in the output/display technology. On the input side, there are far less clear signals on where things will go. Until recently, the standard input equipment for Oculus Rift was still a game console controller. LeapMotion [28], a company known for its free-hand gesture recognition technology, is currently pushing for the embedment of its technology into various VR devices. Microsoft goes to the other end of the spectrum by allowing its mixed reality device HoloLens [30] to accept voice input in addition to some limited gestural input. So, unlike the mouse and keyboard combination we take for granted for desktop computing, the world has yet to see a ubiquitous input mode established for immersive computing. This brings up a series of questions: what are users' preferences, then? Do they always prefer one over another, or is there virtually no difference?

My thesis tries to answer those questions. Before that, thought, this very first chapter will introduce a key concept that will thread throughout my entire inquiry: the type of a task. The

chapter then goes on to list the goals, the questions and the methodology of the research and ends with a preview of all the other chapters.

1.1 Three Distinct Input Modes

Of all the widely available commercial VR/AR products, the input technologies all seem to be different but not *distinctly* different. A closer examination reveals all those technologies are just different combinations of the following three distinct input modes: voice, gesture and controller (Figure 1). For example, the Oculus Touch is a combination of the gesture input and the controller input [29]. Microsoft HoloLens takes advantage of the hybrid use of voice and gesture [30]. Leap Motion's hardware is a pure gesture input device [28].

In the research community, studies have also been done to examine those three distinct modes of input. Many implementation studies have compared user experiences between any two of the three input modes but there is not a clear-cut answer to the question of which input mode is a user's first choice. Either ambiguity exists in a single study [2,6] or there are contradictory claims among different ones [1,10].



Figure 1 - Three distinct input modes: voice (left), gesture (middle), controller (right).
Images from (left to right): <https://www.voicebot.ai>, <http://www.gestigon.com>,
<https://www.engadget.com>

It is no coincidence that the absence of consensus on immersive computing input mode in the research community mirrors the lack of a ubiquitous input mode adopted by industry players in

the commercial world. It seems to suggest there is no “one size fits all” input mode in an immersive computing environment.

1.2 Specification and Manipulation Tasks

If there is no universally preferable input mode for immersive computing, one must ask: what makes a user pick one mode of input over another mode at this moment while she makes a difference choice at the next moment?

In searching for an answer to that, I stumbled upon a research conducted by Morris from Microsoft Research [14] where the investigator was interested in discovering common gestures and speeches adopted by users when they used a web browser in front of a TV. Morris made an interesting observation in her paper:

“...some referents may be best mapped to certain modalities.”¹

What is the “some”? Morris did not give an answer. But at least she suggested *some type* of tasks would be better executed with a specific input mode. A natural follow-up question is: what are the types of tasks, then?

To sort out the types of *all* tasks, however, would be a seemingly intractable task by itself. If I narrow the task domain only to that of visual analytics, though, I can rely on the task taxonomy introduced by Heer *et al.* [8], in which there are three broad types of tasks:

- Data & View Specification
- View Manipulation
- Process & Provenance

¹ “referents” refer to tasks and “modalities” are input modes.

A detailed explanation on them will be spelled out in Chapter 2. For now, briefly speaking, Data & View Specification (referred to as “specification” hereafter for brevity) tasks are mostly about exploring large data sets. View Manipulation (referred to as “manipulation” hereafter for brevity) tasks are largely about drilling down for more details.

With those two types of tasks in place, this thesis is now ready to tackle the question of “why no one-size-fits-all input mode in immersive computing” from a fresh perspective: specification tasks vs. manipulation tasks.

Readers may wonder why I did not include Process & Provenance type of tasks. The reason is that they are not specific to the visual analytics domain. A photo or text editing program may also provide this type of tasks. In addition, Process & Provenance tasks are not essential to conduct visual analytics [8]. Because the premise of this thesis is strictly in the domain of visual analytics, Process & Provenance tasks will not be considered in this study.

1.3 Research Goals

The primary goal of the research is to inform the designers and the developers of an AR/VR project about what input mode they should choose and optimize for users. The secondary goal of the research is to conduct the study in such a way that the results will more reflect users’ thoughts rather than serve as a proxy indicator on the quality of a technology implementation. This is attempted through the employment of a methodology called elicitation study, which will be explained later. The tertiary goal is to bring readers up to date on the latest development in immersive computing interactions within the research community. An extensive literature review will be conducted to achieve that goal.

1.4 Research Questions

Given that we have a way to break down analytics tasks by two types, an immediate question is: will one type of tasks associate more to a specific input mode? In other words,

RQ1: Is there an association between a user's preferred input mode and the type of the task she performs?

Depending on the answer, it could serve as a general guidance for immersive analytics application designers when they need to decide which input mode to use.

Another area this study is interested in is interaction agreement rates². Previous studies [4,12,13,15,17,18,19,20] have shed some light on this but none of them looked at the issue from the *specification versus manipulation* task types perspective. Thus, the following is posted as the second research question:

RQ2: Are interaction agreement rates for one type of task higher than those of the other type?

Answering these two questions will serve the purpose of achieving the primary goal of this research: providing design guidelines for AR/VR developers.

1.5 Research Methodology

Because one of the research goals is to make the results reflect users' ideas, it was decided that the study should not rely on a particular implementation of input technologies. This makes sure that, during the data gathering stage, a study participant's perceived "good" or "bad" of an input mode will not be tainted by the user experience based on a specific implementation of

² Throughout this thesis, whenever we discuss agreement rates, interaction refers to a series of human actions in a specific input mode to complete a task.

an approach nor on its current practical limitations. For example, current voice recognition systems often struggle with non-native accents. Negative interaction experiences resulting from such limits could bias a user's preference. Instead, I opt for the elicitation methodology, which lets a user freely propose whatever interactions she wants, without any consideration on current technology limitations, as long as those interactions make sense to her. There have been a few other studies conducted in such a form to understand users' preferences and interaction agreements in gestural interactions [4, 12, 13, 17, 18, 19, 20, 25, 26]. Detailed setup and procedures of the study will be presented in Chapter 3.

1.6 Contributions

The main contributions of this thesis consist of the following:

1. Discovery of a non-statistically-significant association between a user's preferred input mode and the type of a performed task in most cases.
2. Reveal of significant difference on input agreement rates between two task types.
3. Proposal of an alternative way of quantifying interactions agreements in user elicitation studies.

The first two have more practical implications and are of general interest to someone implementing an input technology. The last one is more foundational and hopefully will generate further discussions on the same topic within the research community.

1.7 Thesis Structure

The current chapter, *Chapter One: Introduction*, has introduced readers to both the central topic of the thesis - input modes in immersive computing - and a key concept threading throughout the entire thesis – the type of a task.

Chapter Two: Background and Related Work – gives a brief overview of the immersive computing technology and surveys the related prior research on input preferences as well as agreement rates in the immersive computing environment.

Chapter Three: Elicitation Study Methodology – details the design and the hardware/software setup for my elicitation study. It also lists the exact steps needed to carry out the experiment and provides an aggregated view of all the participants in the study.

Chapter Four: Elicitation Results – dives deep on how the raw data gathered from the study participants is converted into insights (aggregated data) that are then used to answer my two research questions. It is also in this chapter where an alternative quantitative method on characterizing user interaction agreements is introduced.

Chapter Five: Discussion on Results – interprets the insights derived in Chapter Four and draws conclusions on my two research questions. It provides possible explanations for the observed data in agreements, offers a design guideline for AR interaction designers and reflects on the roles of task types in understanding user inputs for immersive computing. In addition, it explores the practical implication of the alternative formula to calculate interaction agreements.

Chapter Six: Limitation, Future Work and Conclusion – starts with an acknowledgment of several limitations in the study, suggests future work meant to tackle some of the open questions and ends with a summary of the entire thesis.

Chapter 2 Background and Related Work

This chapter starts with a brief overview of the technologies that enable immersive computing, with a focus on the input technology. The purpose of it is to provide readers with the context under which the impact of the thesis work could be readily appreciated, and the possibility of application of the study findings could be assessed, given the relatively abstract nature of this study. The chapter then goes in detail over several prior works from the research community that correspond closely to the two core topics from my two research questions: input preferences and interaction agreements. Because task type is the central theme of my thesis, the last section will be devoted to two literatures that provided a great of inspirations for my study.

2.1 Brief Overview of Input Technologies for Immersive Computing

In late 1980s, VPL Research developed a full body suit equipped with sensors as well as a head mounted display as a pioneering VR product (Figure 2, left). Aside from the huge display, the other impressive components were the gloves used to recognize user's manipulation intents (Figure 2, right).

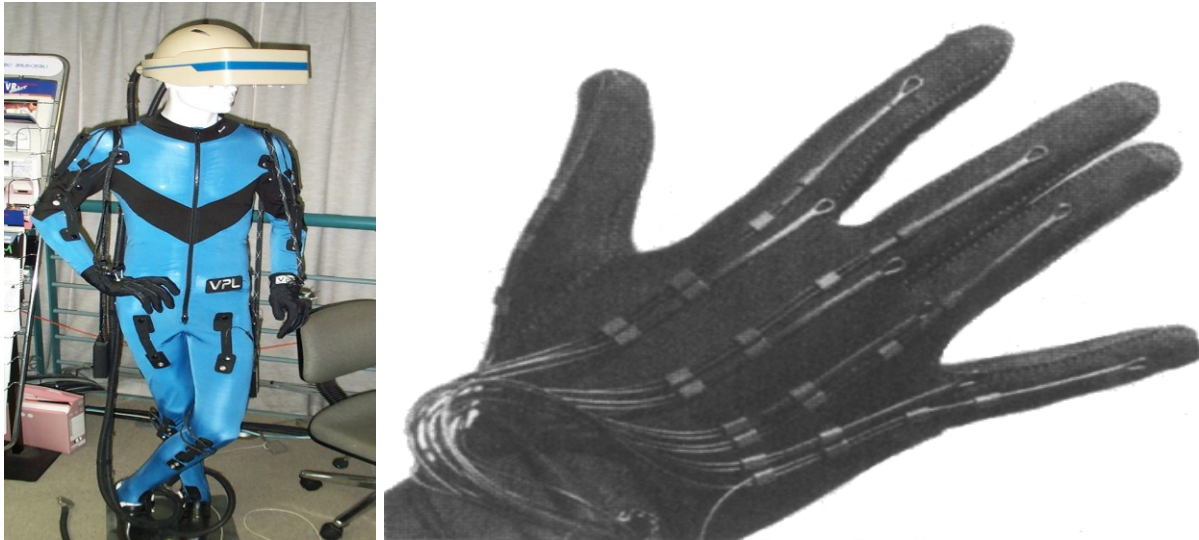


Figure 2 - left: an outfit with body sensors, data gloves and a head mount display; right: a data glove. Images from (left to right): <http://theconversation.com>, <https://www.timetoast.com>

The data gloves offered users 5 degrees of freedom (with ultrasonics) or 6 degrees of freedom (with magnetic flux sensors) and could recognize simple hand gestures [21]. The list of capabilities included finger bending detection, hand tracking and tactile feedback, allowing a glove wearer to manipulate virtual objects rendered in the head mounted display.

In early 1990s, a cubic immersive computing environment, CAVE (Figure 3, left), was created by a group of researchers from the University of Illinois [22]. Users put on custom-made stereoscopic glasses to experience the illusion of being in a lifelike environment. This immersive computing environment did not come with a standard input device. As a result, a game console controller, an air mouse and a 6 degree of freedom wand (Figure 3, right) had all been utilized as an input device for the CAVE [23].



Figure 3 - left: a user interacting with CAVE; right: a wireless wand used as the input device for CAVE. Images from (left to right): <https://www.lifehack.org>, <https://cosmosmagazine.com>

The main difference between the game controller and the wand was that the latter was also able to track a user's hand position with its infrared sensors.

The United States military pushed the envelope on the VR technology as well. The government institution made highly customized, single-purpose input devices for the VR computers used in its training programs. Those VR devices were basically models of the real weaponry (Figure 4).



Figure 4 - left: a military parachuter in VR training environment, using parachute straps as the input device; right: a user in shooting training session with a model gun. Images from (left to right): <http://www.vrs.org.uk>, <https://en.wikipedia.org>

Fast forward to 2010, Oculus Rift was released to the mass market, at first using Xbox game console controller as its input device and then, 7 years later, the newly designed Oculus Touch controllers were introduced as its default input device (Figure 5).



Figure 5 - Oculus Rift Touch controllers. Images from (left to right): <https://www.oculus.com/>

The Touch controllers not only sport conventional buttons, they are also capable of detecting hand rotation and recognizing finger gestures.

In the meantime, Leap Motion offers a free-hand tracking solution for immersive computing. Its USB-connected device can be mounted on a VR device to enable hand gesture recognition in a small area as well as be used in a desktop environment (Figure 6).



Figure 6 - the Leap Motion USB-like device. Left: mounted on a VR device; Right: used in a desktop environment. Images from (left to right): <https://en.wikipedia.org>, <http://smartgimmick.com>

The device constantly records pictures of users' hands and sends the data back to the host computer to process various kinds of hand gestures with a proprietary algorithm.

Recently Microsoft has also joined the immersive computing scene by introducing an advanced untethered AR device named Microsoft HoloLens (Figure 7).



Figure 7 - Microsoft HoloLens, with voice and gesture recognition built in. Right: a HoloLens-wearing user issuing gesture commands in front of mid-air hologram charts. Images from (left to right): <https://www.microsoft.com/en-ca/hololens>, <https://sg.news.yahoo.com/>

It is a full-fledged computer able to provide users with a hologram view, where everything you would normally see on a desktop display is layered over the surrounding physical space. The device can take both voice and gestures as its input, harnessing the same technology Microsoft

has developed in the past with its deprecated Kinect. My study used a Microsoft HoloLens to provide users with an immersive computing environment, too.

2.2 Input Mode Preference

In the research community, study on immersive computing has been going on for quite a while.

Cabral *et al.* built a gesture recognition system to study the usability of gesture interface [2]. Two computing environments were used in the study: one was the immersive environment CAVE; the other was simply a projector screen on the wall (Figure 8).

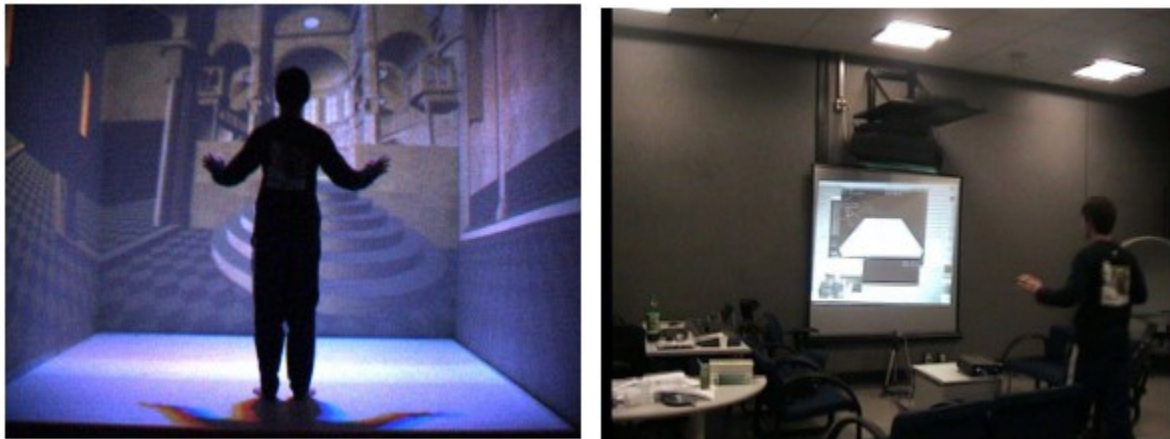


Figure 8 – Cabral *et al.*'s Gesture Usability Study. On the left: gesturing in CAVE; On the right: gesturing in front of a projector screen. Images from: [2]

The first experiment in the study asked the participants to point and click targets appearing in random order, with two different input modes: gesture and mouse. The results showed a significant advantage with the usage of mouse: with it, people were able to complete the random pointing task almost 4 times as fast as they were with gestures. The authors of the study attributed this partly to the lack of gesture experience on their participants' part, but they suggested the range of motions might also be at play here: to reach across a screen, a participant

may only need to move the mouse 1 or 2 centimeters on a hard surface while she may have to move one of her arms dozens of centimeters in the air. In the end, though, the study did not arrive at a conclusive result: while gestures did have advantages such as being natural and intuitive, they slowed down the task completion time and induced fatigue.

Mota *et al.* elicited user feedback on gestures and controllers after letting users explore 3D volume data in a VR setting [6]. The authors developed an oil well exploration application for an immersive computing environment with an Oculus Rift. On the input side, two devices were employed: one was a gamepad with infrared tracking; the other was LeapMotion gesture sensor (Figure 9).



Figure 9 – Mota *et al.*'s study comparing gamepad and gesture inputs. On the left: user with a gamepad; On the right: user interacts with an Oculus Rift where a LeapMotion sensor was attached. Images from: [6]

The study reported that, overall, the study participants preferred gestures over gamepads because gestures felt more natural and intuitive. However, one participant also complained that she must hold her hand above a fixed position in order to have her gestures recognized by the LeapMotion sensor. Another participant in the study pointed out moving across a large region in an oil well with hand gesturing could easily cause fatigues and suggested a remedy of putting one's arm on

the armrest of a chair. Note that this fatigue problem corroborates with the findings from Cabral *et al.*'s paper mentioned earlier [2].

Pick *et al.* conducted a comparison study between speech input and menu-based point-and-click input in a CAVE-like VR setting [3]. The task was layout planning of a factory floor. The menu-based input was done specifically through a hierarchical pie menu system, which the authors developed and then integrated with a layout planning application. To recognize speech, they used a wireless microphone, which sent the captured voice signals to the Microsoft Speech API for further processing. Its speech input mode was not a pure one, though, because it also allowed a pointing device to be used at the same time when a participant uttered voice commands (Figure 10).



Figure 10 – Pick *et al.*'s study where a participant was planning the layout of a factory floor in the CAVE immersive environment. Images from: [3]

In addition to that, the study only allowed for a very specific voice command structure: *Verb, Object[, Adverbs]*. To evaluate the effectiveness of point-and-click input and the speech input,

the study authors deliberately asked their participants to go to the deep end of a 4-level hierarchical menu system. The results revealed that using speech, people were able to complete tasks faster, but more errors could pop up because of either humans' mistakes or flaws in the machines, as compared to the point-and-click input mode. In terms of user experience, the study found neither mode was advantageous.

2.3 Elicitation Studies

All the studies cited earlier had one thing in common: the use of an implemented input system. Therefore, the preference data collected in those studies had a dependency on the performance of the implementations as a user is unlikely to give a high rating to a bug-ridden input system.

If the goal is to simply ask a user, “what input mode do you prefer to do this task?”, instead of “what input mode do you prefer to do this task, *when you use our system?*”, then a study without the use of any implemented input system would seem more appropriate, and that would be an elicitation study.

Elicitation studies arise from the need to maximize the *guessability* of an input system – the chance of a spontaneous user input being recognized by the system. According to Good *et al.*, there are two different philosophical approaches to HCI design [24]:

1. Adapt the user to the system;
2. Adapt the system to the user.

The first is designer-driven, with a hidden assumption that users are always “problematic” and thus need to be “trained”. Because taking the designer-driven approach is basically a form of only improving the *usability* of a system, Good *et al.*, in their quest to build an email system

interface with a high guessability in the early 1980s, took the more user-centric second approach which assumes novice-behavior is inherently sensible. In their study, any spontaneous but unrecognised user input (typed commands) would be intercepted by a human operator, who was unknown to the user but would interpret the command as much as she could to allow a corresponding change to take effect. In the end, all those commands would be incorporated into a new version of the email interface system. After 30 such iterations, the success rate of the system executing spontaneous user commands jumped from a mere 7 percent to an astounding 76 percent [24], dramatically improving the guessability of the email user interface.

Many more recent studies have followed in similar fashion to assess the agreement rate of user inputs. The agreement rate is closely related to guessability [15] and is also one of the central questions of my thesis. The following section will look at some of the other elicitation studies examining agreement rates. First, though, we need to see how the agreement rate was defined by some prior research.

2.4 Input Agreement

2.4.1 Quantification methods.

Typically, in a user elicitation study on input interactions, researchers ask participants to come up with (i.e., propose) several different interactions to complete a task. Later, researchers try to identify the same interactions, by first encoding all the interactions with concise and descriptive texts and then comparing those texts. There is some flexibility in the encoding step, meaning that two visually or verbally different interactions could have the same encoding, as demonstrated in other studies [4,12,18,19,20]. Two interactions are considered the same if they share the same encoding.

Wobbrock *et al.* introduced a formula quantifying user interaction agreement [15] which has been used by several elicitation studies [4,12,18,19,20] (Equation 1).

$$A(r) = \sum_{P_i \subseteq P} \left(\frac{|P_i|}{|P|} \right)^2$$

Equation 1 – Wobbrock *et al.*'s formula [15] for input agreement on a particular referent (task), where P is the set of proposed interactions for the referent, and Pi is a subset of identical interactions from P.

According to the formula, if there are a total of 5 proposed input interactions to complete a task and among them, 2 are considered the same and the other 3 are also considered the same, then

we have $|P_1| = 2$, $|P_2| = 3$ and $|P| = 5$. Thus, the agreement rate for that task $A(r) =$

$$\left(\frac{|P_1|}{|P|} \right)^2 + \left(\frac{|P_2|}{|P|} \right)^2 = \left(\frac{2}{5} \right)^2 + \left(\frac{3}{5} \right)^2 = 0.52. \text{ However, this formula does not account for the case}$$

where none of the proposed input interactions is the same as another. In that case, the agreement rate should intuitively be 0 but according to Wobbrock *et al.*'s formula, we would have $|P_1| = 1$, $|P_2| = 1, \dots, |P_n| = 1$, thus $A(r) = \frac{1}{n} \neq 0$. In other words, if we have a case where two proposals are completely different, we would still end up with a reality-distorting agreement rate of 50%. The error stems from a misperception and an imprecise definition on the concept of agreement, which I will dive into later in Chapter 4.

Taking inspiration from Findlater *et al.*'s alternative agreement measure [27], Vatavu *et al.* fixed the “never 0” problem in Wobbrock *et al.*'s formula by proposing the following equation (Equation 2) [16]:

$$\mathcal{AR}(r) = \frac{|P|}{|P| - 1} \sum_{P_i \subseteq P} \left(\frac{|P_i|}{|P|} \right)^2 - \frac{1}{|P| - 1}$$

Equation 2 – Vatavu *et al.*'s formula [27], an improved version of Wobbrock *et al.*'s formula [15]

Equation 2 has an inclusive range [0..1] for its values, fitting well with one's intuition about agreement. Vatavu *et al.* stated their formula did not immediately invalidate the results of all the previous studies which had relied upon Wobbrock *et al.*'s formula, because the relative orders of agreement rates derived from the two were still preserved (i.e., if $A(r1) < A(r2)$, then $AR(r1) < AR(r2)$, and vice versa). This formula is inapplicable, however, when the number of proposed interactions from a participant is more than 1. It assumes there is a 1-1 mapping between a participant and the proposals (i.e., interactions) she makes. Here is a sentence from the paper making the assumption explicit [16]:

“(two correcting factors in the formula) depend on the number of participants or, equivalently, the number of elicited proposals”

To solve this problem, Morris came up with a metric called “max consensus” [17], defined as:

“the percent of participants suggesting the most popular proposed interaction for a given referent”

The metric does account for the situation where a participant proposed multiple interactions to carry out a task, but the very fact that it utilizes the percent of “participants” rather than that of “agreements” means this metric suffers from the same “never 0” problem as mentioned earlier. Later in Chapter 4, an effort will be made to combine the “max consensus” concept with an idea similar to Findlater *et al.*'s alternative agreement measure [27], to obtain a formula that will provide us with a better understanding on the input agreement among participants who propose multiple interactions for a task.

2.4.2 Empirical studies.

There are studies where agreements were measured based on the formulas mentioned above.

Vatavu *et al.* did some interesting work comparing agreements between handheld gestures and freehand ones [20]. The study asked users to perform typical home entertainment (TV) tasks, including “play”, “mute” and “resize”. Those tasks were further put into three categories: screen-related, function-related and generic. The main devices used in the study were a Wii controller, to allow for handheld gestures, and a Kinect to respond to freehand gestures (Figure 11).

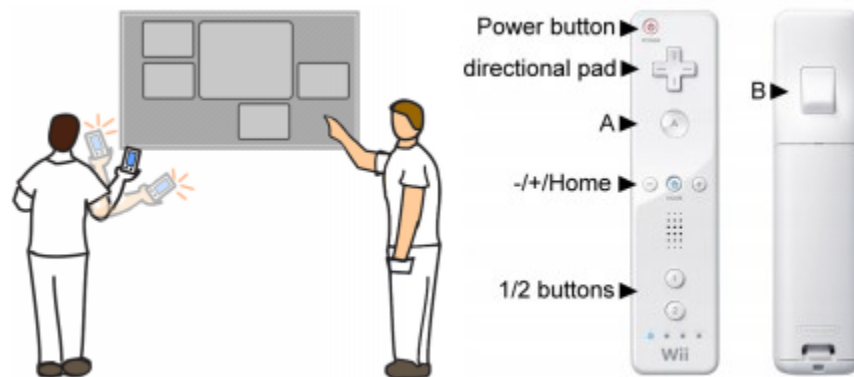


Figure 11 – Vatavu *et al.*’s freehand vs. handheld gestures study. Left: an illustration demonstrating how the gestures were acquired. Right: Wii Remote, the handheld device used in the study. Users were allowed to combine both the button-pressings and the motion gestures together. Images from: [20]

Because it was an elicitation study, no gestures were actually recognized by any computing device. Instead, the study participants were shown animations depicting the effect of a task, and then asked to do a gesture, all the while imagining the gesture would be recognized to complete the task. The analysis on gesture agreement from the study was based on Wobbrock *et al.*’s formula (Equation 1). It showed there was no statistically significant difference between

agreements of handheld gestures and freehand gestures, but the study stopped short of comparing agreements among the three task categories it introduced.

Piumsomboon *et al.* simulated an AR environment for an elicitation study where the participants were asked to come up with gestures for a broad range of common but rudimentary tasks [12], of which most would be considered as specification tasks by this thesis, such as “copy”, “accept” and “stop”. The combined use of a head mounted display (HMD) and a camera allowed the study participants to have a simulated AR experience (Figure 12).

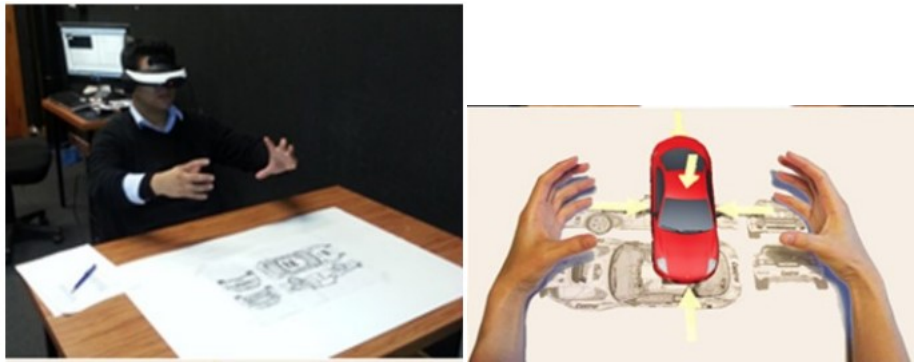


Figure 12 – Piumsomboon *et al.*'s gesture study with an AR simulation. Left: a participant in the study; Right: the simulated AR experience from a participant's view, with the car visual provided by the HMD and the table view provided by the camera attached to the HMD. Images from: [12]

Like most other gesture elicitation studies, the participants first saw an animation to understand the task she was asked to do and then came up with a 1-hand or 2-hand gesture to carry it out. No gestures recognition software was used, and the participants were told to put behind them the worries about technology limitations. As it turned out, in this 20-person study, three tasks in the study, “previous”, “next” and “select from a vertical menu”, achieved a perfect agreement rate of 1. Note that “previous” and “next” would be classified as manipulation tasks according the specification/manipulation taxonomy introduced earlier in Chapter 1.

Morris, employing the “Wizard of Oz” methodology [5], elicited gestures and voice commands for using a web browser on a TV [17]. The basic setup was a wall-mounted 63-inch TV with a Kinect placed on top of it. In this case, the Kinect functioned purely as a distraction so that the authors could act as a hidden “wizard” behind the scenes (Figure 13).



Figure 13 - The living room used in Morris’ *Web on the Wall* study. Note there is a decoy Kinect placed on the TV. Images from: [17]

Unlike the previously mentioned elicitation studies where there was only one participant in a single round of elicitation, there were two or three participants jointly present in a round. The participants were asked to carry out a series of predetermined tasks, with gestures and/or voices, to plan a weekend activity together using a web browser. The tasks included “open browser”, “enter URL” and “reload page”. The participants were made to believe that the Kinect they saw on top of the TV was functioning and able to recognize reasonable gestures and voice commands. In reality, though, the authors went behind the scenes to remote control the browser so that they could react properly to the gestures and the voices made by the participants. The participants were also allowed to come up with multiple interactions for a single task, which led the authors to introduce the concept of “max-consensus” to calculate the gestures/ voices

agreement rates with Webbrock *et al.*'s formula. After comparing agreements rates between gestures and voice commands, the study found gestures, on average, had a higher agreement than voice commands but the difference was not statistically significant. Further analysis on a per-task basis, however, revealed significant agreement differences between the two input modes in some tasks. As a result, the study suggested some tasks were better suited to a specific input mode. But the study did not go further to exam the characteristics of those tasks.

Kühnel *et al.* did a gesture elicitation study in the field of smart home control [4]. What is notable about the study, particularly through the lens of my own study, is that it measured gesture agreements by gesture types. In Kühnel *et al.*'s study, there were four types of gestures: physical, metaphorical, abstract and symbolic, which seemed more fine-grained than the binary classification provided by Quek *et al.* [9]. What the study participants were asked to do was to perform gestures with an iPhone to control home devices such as a TV, lamps and a blind (Figure 14).

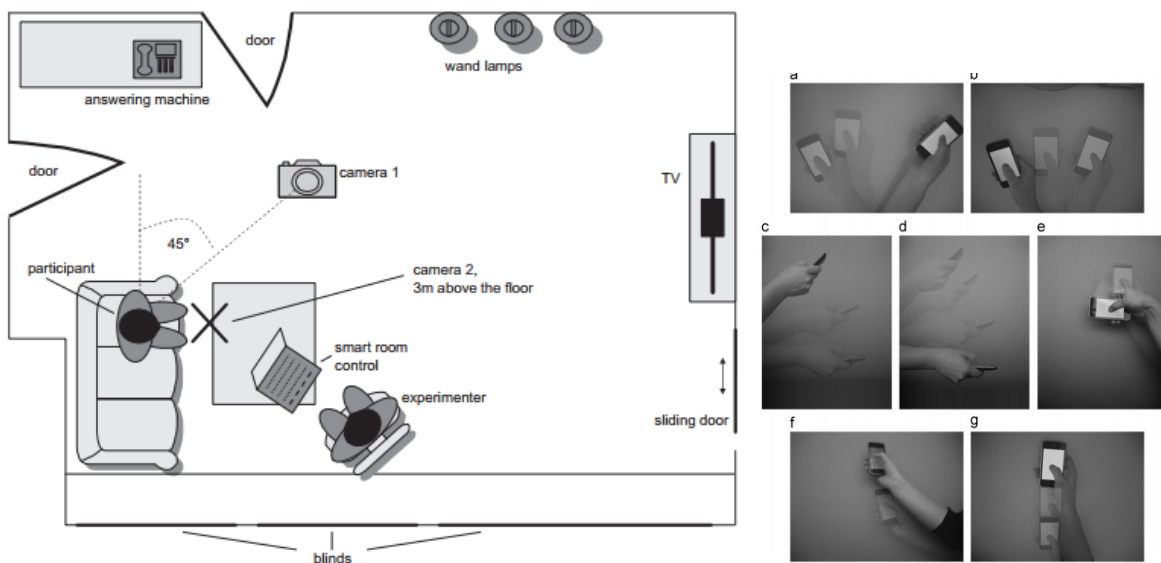


Figure 14 – Kühnel *et al.*'s smart home study. Left: the lab setup. Right: different gestures proposed by a participant in the study. Images from: [4]

The tasks included “roll up blind”, “dim the lamp” and “switch to next channel”. Like the elicitation studies cited earlier, no attempt was made to recognize gestures at all. The authors instead used a laptop to control the smart home devices, showing the “effects” to a study participant. The participant was then asked to come up with whatever gestures she deemed appropriate to achieve that same effect. At the end of each task, the participant would provide a subject rating on how suitable and easy it was to perform her proposed gesture. My study follows a similar procedure. After an extensive analysis, Kühnel *et al.* found out that the type of gesture with the highest number of agreements was “physical” while the one with the lowest agreements was “abstract”. Unfortunately, neither type was given a precise definition, so it is hard to relate Heer *et al.*’s “specification” task type with theirs. In addition, it seems there were some similarities between “physical” and “metaphorical” types, and some similarities between “abstract” and “symbolic” types. Again, without precise definitions, it is hard to make a conclusion. Having said that, the study also took an interesting perspective by looking at gesture agreements through the lens of task complexity. The complexity of a task was perceived by the authors to be the difficulty of performing the task. The conclusion they drew was that the more complex a task was, the lower the gesture agreement for it was. Once again, the authors did not spell out exactly what constituted as “difficult”. Otherwise it would be interesting to see how the idea of manipulation/specification tasks introduced by Heer *et al.* could relate to their concept of task complexity.

2.5 Task Taxonomies

In the visual analytics domain, there are some studies on the classification of analytics tasks. Two of them were particularly relevant to this study.

The first one [7], by Amar *et al.*, articulated that many of the existing task taxonomy provided by the information visualization research community were too system-oriented. They were all based on the specific functionalities offered by visualization software. Amar *et al.* wanted to flip this process by asking users to come up with a set of data analysis questions and then extracting the “core desire” from those questions to form a list of tasks, upon which a new set of classifications would be generated. Note this approach also fits in with the previously mentioned “adapt the system to the user” design philosophy, a more user-centric methodology that often manifests itself in the form of user elicitations. Amar *et al.* asked their students to generate a series of analytical questions with regard to some data sets. They then grouped those questions by similarity as well as the so-called “core knowledge goal” of the questions. In the end, they produced an analytic task taxonomy consisting of ten types of tasks:

- Retrieve Value
- Filter
- Compute Derived Value
- Find Extremum
- Sort
- Determine Range
- Characterize Distribution
- Find Anomalies
- Cluster
- Correlate

Amar *et al.* considered those tasks as “primitives” and thought it would allow for an easy reasoning about compound tasks. They did acknowledge that during the taxonomy producing

stage, some questions were omitted, either because the questions were too mathematically primitive or because the questions were subjective in the sense that they involved unspecified value judgements.

The other literature [8], in the form of a journal article written by Heer *et al.*, which was already mentioned in the previous chapter, proposed a visual analytics task taxonomy that was largely based on prevalent functions provided by visualization software. The taxonomy was made up by 12 types of task grouped into three broad categories, shown as follows:

1. *Data and View Specification*

- Visualize
- Filter
- Sort
- Derive

2. *View Manipulation*

- Select
- Navigate
- Coordinate
- Organize

3. *Process and Provenance*

- Record
- Annotate
- Share
- Guide

According to Heer *et al.*, Data and View Specification tasks allows users to explore large sets of data by *visualizing*, *filtering* and *sorting* existing data as well as *deriving* new data from existing one. Examples of deriving include normalizing values, running statistics and aggregating data.

View Manipulation tasks, on the other hand, allows users to highlight patterns and drill down for more fine-grained details by *selecting* items or data regions, *navigating* views (scroll, pan, zoom etc.), *coordinating* as well as *organizing* among multiple views. One example of coordinating, according to Heer *et al.*, is “selecting items in one display to highlight (or hide) the corresponding data in the other views”. Organizing refers to do the proper layout of multiple views. While users can manually arrange multiple views to arrive at a suitable layout, Heer *et al.* suggested those tasks ought to be automated intelligently by software.

The last category of tasks, Process and Provenance, enables users to do iterative data exploration and interpretation through *recording*, *annotating*, *sharing* and *guiding*. Recording, according to Heer *et al.*, allows for undo and redo. Annotating, in the form of freeform graphical markings on a view, is used to communicate insights about data. An example of sharing is turning visualization dashboards into interactive web pages. Guiding is for the visualization software to provide hints, explanations, or even tutorials along a user’s data exploration process.

Those two taxonomies, especially the first two task categories from Heer *et al.*’s classification, proved to be very useful in the set-up of my elicitation study, which will be explained in detail in the next chapter.

2.6 Where My Study Differs

Compared to all the studies on preferences and agreements cited in this chapter, my study differs mainly in the following two ways:

- First, unlike those previous work that studied and compared between only two input modes, this study exams preferences and interaction agreements across three distinct input modes: gesture, voice and handheld controllers.
- Second, those studies did not investigate the input mode preferences or the interaction agreements by the types of the underlying performed tasks. This study employs a fresh perspective of specification tasks vs. manipulation tasks.

Chapter 3 Elicitation Study Methodology

To yield answers to my two research questions, an empirical study was conducted. As one of my goals was to determine user preferences, I followed the protocols from other similar studies to use an elicitation approach to gather all the data needed for my investigation. On the highest level, my study involved asking participants to do tasks through hand gestures, voice commands and handheld controllers without being constrained by respective technology capabilities. Two types of data were collected:

1. Participants' explicitly stated preferences for an input mode;
2. Recordings of participants' interactions.

As mentioned before, this study intentionally does not use an implemented recognition system for either gestures, voices or controllers. On one hand, this deliberate non-use of an implemented system makes the study a bit abstract, but on the other hand, the completely user-centric approach ensures the conclusions drawn are much more generalizable as they are independent of any limitations innate in specific implementations.

3.1 Terminology: Referent vs. Task

Some elicitation studies [4, 13, 17, 18, 20] have used the term “referent” to refer to the actions they asked users to perform, which could be instances of term misuse. They all cited Wobbrock *et al.*'s surface computing gestures study [19] when the term first appeared in each of their own studies. In Wobbrock *et al.*'s study, “referent” was defined as “*effects of an action*” (but not the action *itself*).

Those later studies, however, continued to use the term “referent” to refer to *actions* such as “record movie” [4], “find” [13], “click link” [17], “send” [18] and “pause” [20] without elaboration.

Therefore, this thesis is going to follow Piumsomboon *et al.*’s example [12] by adopting a more plain-sounding but also more accurate term “task” to refer to the actions that the study participants were asked to do.

3.2 Task Selection

This study consulted the previously mentioned analytic task taxonomies developed by Amar *et al.* and Heer *et al.* [7, 8] and used them as a foundation to build up a list of specification and manipulation tasks, as seen in Table 1 and Table 2. Note the actual tasks performed by participants are listed in the “Example Used in Elicitation” column. The “Before Execution” and “After Execution” columns serve as visual descriptions for what each task is. The former shows what the visual state is before a task is performed while the latter shows what the state should be after the task is performed.

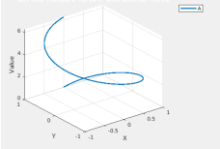
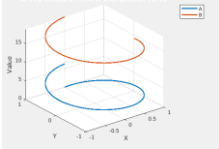
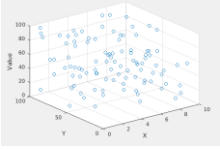
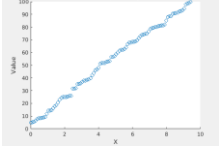
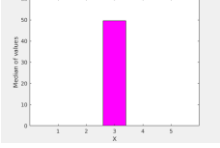
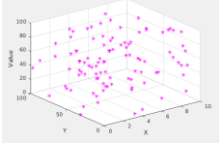
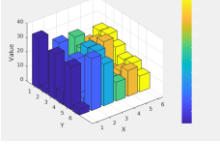
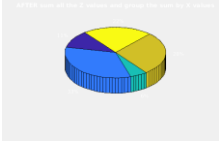
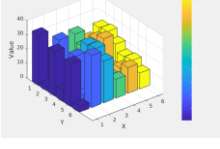
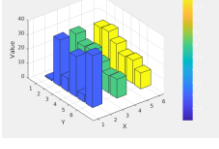
Task	Example Used in Elicitation	Before Execution	After Execution
compare	comparing two curves		
sort	sorting points by Z-axis values		
disaggregate	breaking a median bar to individual points		
aggregate	summing Z-axis, group the sum by X-axis		
filter	filter by even X-axis values		

Table 1 - Specification tasks used in the elicitation

Task	Example Used in Elicitation	Before Execution	After Execution
zoom in	zoom in		
zoom out	zoom out		
single-select	selecting one point on the curve		
multi-select	selecting multiple points		
rotate	rotating horizontal bars		
highlight	highlighting a group of points		
left pan	moving all points to left		
right pan	moving all points to right		

Table 2 - Manipulation tasks used in the elicitation

As for why those specific tasks were used, it was because “Sort” and “Filter” were explicitly listed in both taxonomies. “Compare” was inspired by “Correlate” in Amar *et al.*’s taxonomy. “Aggregate” was mentioned in Amar *et al.*’s article. The reverse - “disaggregate” – was suggested by an expert in the visual analytics. All of them were emblematic of specification tasks. In terms of manipulation tasks, they were largely based on Heer *et al.*’s visualization task taxonomy. “highlight”, “single-select” and “multi-select” were variations of Heer *et al.*’s “Select”. “zoom in/out”, “left/right pan” and “rotate” were concrete examples of “Navigate”.

For each elicitation participant, however, not all tasks were performed. To avoid a noticeable amount of physical strains caused by the weight of the HoloLens on a participant’s head, she would be asked to perform only six tasks out of a total of 13. At first, half of the six tasks were randomly chosen from the specification type and the other half randomly from the manipulation type. But it was later found out after 16 participants had gone through the elicitations that this seemingly even-handed approach had led to an oversampling of specification tasks, because there were more manipulation tasks than specification ones to choose from (8 vs.5). Oversampling of specification tasks could compound any special effect of them on the measured variables (i.e., preference and agreement). To mitigate this issue, it was decided to bring more participants to do manipulation tasks, so that the ratio of performed specification tasks to performed manipulation tasks would closely match the 5:8 ratio. Therefore, after running 16 participants on the 3-specification-3-manipulation format, another 5 were added, each of whom did 6 manipulation tasks.

3.3 Apparatus

The main device used in the elicitation study was a Microsoft HoloLens. As mentioned in Chapter 2, HoloLens, unlike virtual reality devices such as Oculus Rift, makes a user see virtual objects as holograms floating in mid-air, thus allowing her to view the physical space around her as well.

For the study, what a participant would see after putting on the HoloLens are exactly those 3D perspective images listed in Table 1 and Table 2, except for the background which was changed to be transparent in order to maximize one's immersive feeling. Figure 15, for example, shows what a participant would see if she was doing the “zoom in” task.

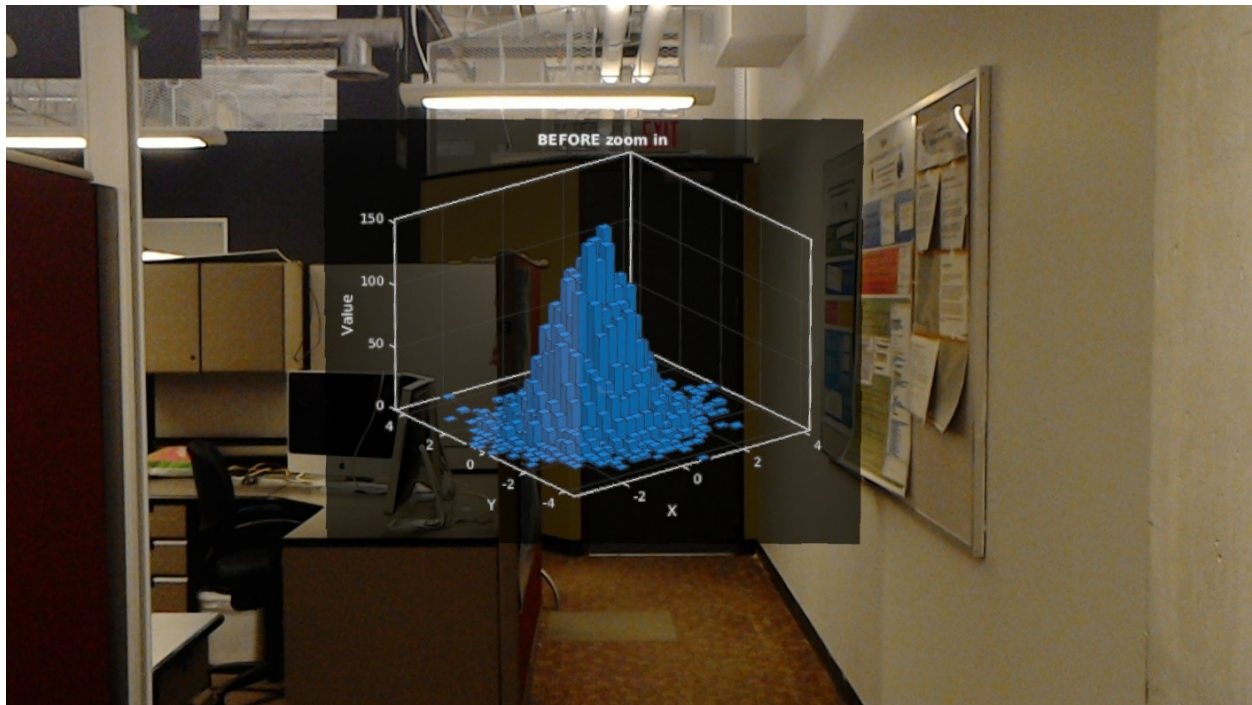


Figure 15 - Screenshot of what a participant would see in HoloLens.

All those “before” and “after” images were generated with custom MATLAB code.

The other device used by participants was a generic game console controller resembling an Xbox controller. The reason for not using an AR/VR controller like the ones paired with

Oculus Rift (i.e., Oculus Touch) is that their usage involves a mixture of gestures and button pressing. That kind of bimodal device would have blurred the results I was seeking from the three distinct modes of input: voice, gestures and controllers, each of which is clearly unimodal. Figure 16 shows an Xbox controller. The generic controller used in the elicitation shares the same layout.

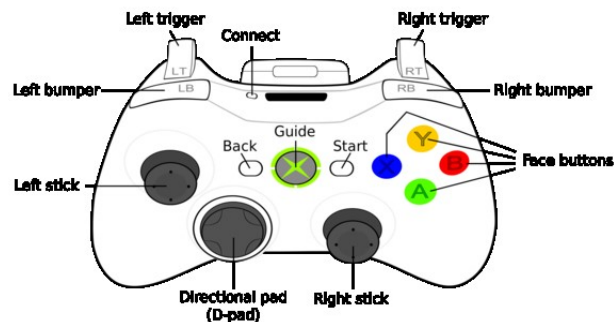


Figure 16 - Button layout of an Xbox controller. Image from:

<https://commons.wikimedia.org>

Aside from those essential equipment, a chair was also in place to be sit on by a participant. A camera was situated approximately 5 meters in front the chair to record the participant's interactions.

3.4 Procedure

First, the purpose of the study as well as what the elicitation would entail was described to a participant. After consenting to be video recorded for the elicitation, she was asked a series of questions about her level of experience with gesture interactions, voice commands and game controllers. The participant's age was noted, too.

Next, to make sure the participant fully understood all the tasks she would be asked to do, I would explain to her the desired effect of each task by showing on a laptop both the "before" and the "after" images of the task. Because the participant was supposed to come up with her

own voice commands, I was careful enough in my verbal explanation to avoid the use of imperative statements so as not to predispose her toward a particular voice command.

Finally, the participant would go to a chair and put on the HoloLens and be told there was no recognition system in place and, for gestures and voice commands, she should come up with whatever made sense to her as long as she felt it could express her intent well. For the controller, all she needed to do was to move its sticks and/or press its buttons and, if necessary, imagine some UI elements in a displayed image to suit the point-and-click interaction paradigm of the controller.

For each task, the participant would be shown its “before” image in the Windows Photos app that came with the HoloLens. She would then do a gesture, a voice command or controller movements (Figure 17).



Figure 17 – A study participant doing gesture for the task “single-select”.

To simulate the effect of the participant's action, I would, as soon as she finished it, switch the displayed image to the corresponding "after" image with the click of a mouse. The mouse was connected to the HoloLens through blue-tooth while on my Windows laptop, an app called Microsoft HoloLens provided me with a view mirroring what the participant saw in the HoloLens.

The participant would be asked to come up with as many as three different gestures to complete a task while sitting or standing. The sitting/standing position would alternate between each task. A participant with an even-numbered participant ID would start with the standing position whereas one with an odd id would begin with the sitting position first. The purpose of such an arrangement was to obtain participants' preferences on sitting vs. standing. After gesturing, the participant would issue up to three different voice commands to complete the same task. The last input mode to be evaluated was the controller input, where the participant was required to demonstrate just 1 set of controller movements.

At the end of each task, I would ask the participant to name her preferred input mode for it and the reason for her choice. She would also be asked at the end of the entire elicitation about her preference for sitting vs. standing while doing gestures.

Some readers by this point may wonder why I asked for 3 gestures and 3 voice commands but only for 1 set of controller movements. The reasons for three gestures and voice commands were simply:

1. to reduce legacy bias (from the use of mouse and keyboard) as suggested by Morris *et al.* [14];
2. to maximize the chance of finding similar interactions.

As for controllers, because they are designed for the prevalent “point-and-click” interaction paradigm, there is no “legacy bias” that can be avoided. In addition, unlike gestures or voice commands which allow for unrestricted forms of expression, controllers do have a fixed number of buttons one can choose from, therefore reducing the number of various interactions a participant can come up. Based on those two observations, I concluded the benefit of asking participants to come up with more than 1 set of controller movements was neglectable and thus decided against it.

3.5 Participants

In total, 21 participants joined the elicitation. Eleven were females and ten were males. The youngest participant was 18 while the oldest was 65 (mean age was about 27, median age was 24, with SD about 10). Eight of them categorized their experience with game controllers as “a lot” and another 3 said they had “a lot” of experiences with voice commands. 4 participants indicated “a lot” of experiences with gestures.

In the next chapter, I will go over the data I gathered from all of them.

Chapter 4 Elicitation Results

This chapter shows an aggregation of the raw data collected from the elicitations (the raw data itself can be found in Appendix A and B). I will first present the aggregated data on preferences, followed by quotes from some participants on why they made a particular choice. The chapter will then go on to show the aggregated data on agreements. However, because of the limitations in the existing agreement formulas described earlier, I will take a detour to introduce an alternative one, before revealing the agreement results calculated with my formula.

4.1 Data Size

In total, there were 126 instances of task execution from twenty-one participants, with each of them doing six. Manipulation tasks were executed 78 times while specification tasks were performed 48 times. As mentioned in the previous chapter, this imbalance was intended so that on average, a single manipulation task and a single specification task had about the same times of being executed (for a total of 8 manipulation tasks, it is $\frac{78}{8} \approx 10$ times; for the 5 specification tasks, it is $\frac{48}{5} \approx 10$ times). Because not all participants could come up with three gestures and three voice commands (e.g., two of the participants could not come up with a voice command for the “single-select” task; one participant could not come up with a gesture for the “sort” task, etc.), the elicitation data actually consists of 269 gestures, 271 voice commands and 125 sets of controller movements.

4.2 Input Mode Preference

The preference data collected in the elicitations looks like the following (Figure 18):

subject id	standing/sitting pref. when gesturing	task	type	voice	controller	gesture
12	sitting	aggregate	<u>specification</u>	best		
		compare	<u>specification</u>		best	
		disaggregate	<u>specification</u>		best	
		left pan	<u>manipulation</u>			best
		right pan	<u>manipulation</u>			best
		rotate	<u>manipulation</u>			best

Figure 18 – A snippet of preference data collected in the elicitations.

The value “best” in the “voice”, “controller” and “gesture” columns indicates a participant’s preference with a corresponding input mode.

If we do not discriminate the results by task types, then it shows the participants preferred “gestures” 56 times, “voice” 37 times and “controller” 33 times overall. In relative terms, it is 44% of overall counts for gesture, 29% of counts for voice and 26% of counts for controller.

If we are to break down the results further by task types, we cannot simply limit ourselves to one type of tasks and then calculate a preference rate, because not all tasks have the same number of occurrences due to the random sampling. One way to handle this kind of situation is to simply list preference count for each task. Here is the result:

task	type	preference count		
		voice	gesture	controller
aggregate	specification	7	0	3
compare	specification	5	1	2
disaggregate	specification	3	4	4
filter	specification	9	1	0
sort	specification	7	0	2
highlight	manipulation	1	8	1
right pan	manipulation	0	6	1
multi-select	manipulation	2	4	2
left pan	manipulation	0	12	2
rotate	manipulation	0	6	3
single-select	manipulation	0	3	7
zoom in	manipulation	0	7	2
zoom out	manipulation	3	4	4

Table 3 - Preference count for each input mode in all the 13 tasks

4.2.1 Statistical significance.

To examine the statistical significance of the preference difference among the three input modes for each task, calculations of p-values were needed. In this study, the threshold was set at 0.05. My general strategy was to first examine the difference significance across all the three input modes. If the resulting p-value was smaller than the threshold, then a post hoc analysis involving three additional pairwise comparisons between every input mode would be performed, so that I could find out which difference was significant. If the p-value from across the three input modes no smaller than the threshold, then no post hoc analysis would be conducted.

Since the preference data (preferred vs. non-preferred) was binomial, Cochran's Q test was used to derive the p-value indicating the significance of the preference difference across all the three input modes. Then, if a post hoc analysis was warranted, McNemar's test would be used to calculate a p-value concerning any pair of input modes. Because post hoc analyses would introduce the multi-testing problem, Bonferroni correction was also used to compensate for that by inflating all the p-values resulting from McNemar's test. The following table lists the calculated p-values.

task	type	p-value from Cochran's Q test across 3-input pref.	p-value from McNemar's test with Bonferroni correction		
			between voice-gesture pref.	between voice-controller pref.	between controller-gesture pref.
aggregate	specification	0.02	0.06	1	0.75
compare	specification	0.2	n/a	n/a	n/a
disaggregate	specification	0.91	n/a	n/a	n/a
filter	specification	<0.01	0.09	0.02	1
sort	specification	0.01	0.06	0.54	1
highlight	manipulation	<0.01	0.15	1	0.15
right pan	manipulation	0.01	0.12	1	0.39
multi-select	manipulation	0.6	n/a	n/a	n/a
left pan	manipulation	<0.01	<0.01	1	0.06
rotate	manipulation	0.05	n/a	n/a	n/a
single-select	manipulation	0.02	0.75	0.06	1
zoom in	manipulation	0.01	0.06	1	0.54
zoom out	manipulation	0.91	n/a	n/a	n/a

Table 4 - p-value concerning the difference of preference counts for all tasks

4.2.2 An aggregated view of reasons.

Aside from asking the participants what input mode they preferred, I also asked why they preferred it. To list every response would make the thesis too verbose. Instead, I chose to aggregate the response data by extracting three most common keywords from the responses (a keyword would count only once for a single response even if the word appeared multiple times in the response). Among the preferences for *gesture*, the most common keywords are “precise” (16, 28% of all responses favoring gesture), “easy” (15, 27%) and “intuitive” (8, 15%). For responses favoring *voice*, the most common keywords are “easy” (19, 51% of all such responses), “quick” (6, 16%) and “convenient” (2, 5%). In the group where *controller* was preferred, the top three keywords are “easy” (8, 23%), “precise” (7, 22%) and “familiar” (6, 20%).

4.2.3 A sample view of individual reasons.

Even though not every response is going to be enumerated, there are a few of them that are either thoughtful or interesting and thus deserve a spot here. For instance, when asked about the input mode preference for the task “zoom in”, one participant said this:

“My most favorite mode is gesture for sure...just because it feels natural and intuitive...with voice...I have to say something like ‘hey, try this!’ and then wait for the visual feedback whereas with gesture, I’m manipulating and seeing the results as I move my hands...controller is my least favorite because I have to pick it up and cannot do other things with my hands...also I have to carry it around...even though one benefit with the controller is its tangible feedback...”

Another participant, after finishing the task “single-select”, explained why he favored “controller” over the other two input modes:

“Because I have accurate control...my least favorite is definitely the voice, because I have to very clearly specify which one (point) I want to select, and I cannot do that without a cursor..., (with regard to) gesture, your finger is only so accurate unless you are up close to it (the image)...even if I can bring the image close to me, there may be cases where I do not want to do that. For example, if I’m doing a presentation and we are all seeing the same shared image. I do not want to bring it closer (to select a point) and then put it back to show everyone...every time. Gosh!”

The last quote to be included comes from a participant whose favorite input mode was “voice” when it came to the “aggregate” task:

“Voice...it’s a much saner way to describe a complex idea and you can simply say (it) and mean what you want whereas (with) gesture, you will have to... [the participant tried to

move his hand and then stopped, seemingly searching for words] ... you know, it's a more abstract version (of doing the task) ... (and) it's the worst (input mode)"

4.3 Sitting vs. Standing While Gesturing

Asked whether it is better to sit or stand while gesturing, 52% of the participants answered “sitting”, 29% indicated no preference and the rest 19% chose “standing”. Almost all who preferred sitting cited “comfort” as the primary reason behind the choice. The other reasons are “Device (HoloLens) too heavy” and “Tasks are stationary. No need to stand”. The ones who liked standing thought that the position afforded them more space to move their arms and hands.

4.4 Input Agreement Rates

Finding agreements, by definition, involves comparison. The raw data collected on participants’ gesture, voice and controller interactions was in the form of video recordings. To make the comparison process faster as well as less error-prone, I was advised to encode the video and audio content into texts.

Therefore, for gesture videos, I paid attention to the following details from each participant’s gesture: number of hands used, movements of the hand(s), static hand poses, the actions of active finger(s) and number of repeated movements. With them, I was able to come up with descriptions such as the one below for a gesture performing the “aggregate” task:

(one hand, fingers bunching, moving up and down) x5, palm facing camera and turning clockwise

The “x5” in the encoding indicates the gesture in the brackets was repeated five times.

For the controller input encoding, it was much easier. I only took note of the buttons pressed, clockwise/counter-clockwise moves of the two sticks, which direction button was pressed on the directional pad (D-pad) and the number of repetitions. Here is a controller encoding for the “aggregate” task:

left stick counter-clockwise, right stick counter-clockwise, (yellow button) x2, d-pad up

Voice encoding was even more straightforward as it is simply a transcription of what a participant had said. Here is a voice command issued by a participant to perform the same “aggregate” task:

“add all the values across y-axis together, make a pie chart, distribute it by x”

4.4.1 Definition of same inputs.

The encodings resulting from the above procedure had to go through a consolidation in order to reflect the true intent behind each input. The encodings were admittedly approximations of reality and a rigid comparison of those first phase approximations would only yield a result unable to reveal the real signal. My real goal was to find out how many participants shared the same mental model, or thought alike, when they performed a task. Some gestural studies [12, 13] also loosened their definitions for the “same” in order to capture participants’ real thinking behind their proposed gestures.

In this case, I borrowed the definition for “same gestures” directly from Piumsomboon *et al.*’s study: “gestures that were identical or having consistent directionality although the gesture had been performed with different static hand pose” [12]. For example, these two gestures - “one hand, closed fist, moving back” and “one hand, three-finger grasp, moving back” - would be considered to be the same because they have consistent directionality even though the static hand

poses - “closed fist” and “three-finger grasp” – are different. In practice, this means I would consolidate the two encodings by replacing one with the other.

As for voice, it would be quite a distortion to regard utterances such as “take out Y, sort by values” and “take out Y, sort based on values” as completely different. Instead, I defined “same voice” as “utterances that starts with as synonymous verb or verb phrase, optionally followed by synonymous nouns or attributes, optionally followed by synonymous adverbs”. This means that I had to, by taking out non-essential words like “a” and “the”, trim down each transcript to its bare minimum structure resembling “*verb | verb phrase* [, *noun | attributes*], [*adverbs*]” (“|” denotes “or”, “[...]” denotes optional content), and then consolidate the synonyms by using one consistent word or phrase.

Last, the definition for “same controller inputs” initially was “the exact same sequence of actions (button pressing/D-pad/stick moves)” but I soon found it too limiting. Even though the participants pressed different buttons, most of them were just meant to confirm an action (according to the participants’ own comments). In addition, most of the directional moves employing either the two sticks or the D-pad were arbitrary because the intend was to move the cursor between imaginary UI elements (again, according to the participants’ own comments). Those few non-arbitrary moves, which were related to positional placement of objects on the screen, were still arbitrary from my point of view because I never set an initial position of the cursor for the participant. Given those two observations, I loosened the definition for “same controller inputs”. It regarded all four face buttons (the colored ones as shown in Figure 16, Chapter 3) as the same (unless a participant used multiple face buttons and explicitly named each by color), and all directional moves employing the two sticks or the D-pad as the same. So, an

encoding like “left stick counter-clockwise, right stick counter-clockwise, (yellow button) x2, d-pad up” would be reduced to “stick turn, button x2, d-pad”.

The complete encodings for all the three input modes can be found in Appendix B of the thesis.

4.4.2 Clustering of same inputs.

After consolidating encodings for each of the three input modes, it is a straightforward process to cluster the encodings for a single task simply by checking whether those encodings were the same. Here is an example of such clustering:

- Before clustering: "task 2": [
 - "id 1": ["gesture B", "gesture A"],
 - "id 8": "gesture A",
 - "id 2": "gesture C",
 - "id 3": "gesture A"
]
- After clustering: "task 2": [
 - "gesture C": "id 2",
 - "gesture A": ["id 1", "id 3", "id 8"],
 - "gesture B": "id 1"
]

The “id #” represents a participant ID. Now the question is: how much agreement is there among the four participants when it comes to gesturing for “task 2”?

4.4.3 An alternative formula for agreement rates.

I could have followed the footsteps of many other studies [4, 12, 13, 15, 18, 19, 20, 25, 26] to use Wobbrock *et al.*'s formula (Equation 1, Chapter 2) to calculate agreement rates.

However, as it has been pointed out, Wobbrock *et al.*'s formula distorts reality by never equating to 0.

Vatavu *et al.*'s formula [16] fixed the problem but was inappropriate for my study where multiple input proposals were sought from a participant for each task. To see why, consider a hypothetical situation where one participant proposed one gesture ("id 1": "gesture A") and another participant proposed an infinite amount of gestures, one of which was the same as the one proposed by the first participant ("id 2": ["gesture A", "gesture B", "gesture C", ..., "gesture N"]). Using Vatavu *et al.*'s formula,

$$\mathcal{AR}(r) = \frac{|P|}{|P| - 1} \sum_{P_i \subseteq P} \left(\frac{|P_i|}{|P|} \right)^2 - \frac{1}{|P| - 1}$$

the agreement rate between the two participants would be 0. But this has more to do with the second participant ("id 2") disagreeing *with herself* endlessly rather than a true reflection of the difference between the two people. After all, the first participant ("id 1") only has one idea and the second participant agrees with her totally on that. I call this "agreement paradox". Morris made a similar but more practical example to illustrate the same point [17].

To provide a more fitting solution to the problem, Morris introduced the concept of "max-consensus". The only drawback with that approach is the retainment of the "never 0" problem. To illustrate, let us for a second time imagine there are 2 people each giving a different proposal. Then each of them can say her own proposal is the most popular one. Therefore, the "max-consensus", defined as "the percent of participants suggesting the most popular proposed interaction" [17], is $\frac{1}{2} = 50\%$, a number defying our intuition again.

The following table summarizes the characteristics of the three agreement rate metrics discussed so far:

min value	Never 0	0
<i>applicable study</i>		
<i>Single-proposal</i>	Wobbrock <i>et al.</i> 's formula	Vatavu <i>et al.</i> 's formula
<i>Multi-proposal</i>	Morris' max-consensus	?

Table 5 - Characteristics of the three agreement rate metrics used in prior research

From the table it is clear to see this study needed a metric (marked as “?”) which not only minimizes to zero but also can aggregate the multi-proposal-per-participant data.

To this end, I propose the following formula to calculate agreement rates where a participant makes multiple input proposals to perform a task:

$$AR = \frac{\max_{P_i \subseteq P} |P_i| - 1}{|P| - 1}$$

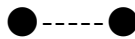
where $|P|$ is the number of participants, P_i is a set of participants who made proposal i , so

$\max_{P_i \subseteq P} |P_i|$ is the number of participants who made the most popular proposal.

This formula can be viewed as a modification on Morris' “max-consensus” as her idea can be simply written as $\frac{\max_{P_i \subseteq P} |P_i|}{|P|}$. The subtraction of one from both the numerator and the

denominator in the formula, in my opinion, gives a more intuitive portrait of “agreement rate”.

To explain it, let us examine the following visualization for two people who agree with each other:



If we chose to describe the magnitude of the agreement between the two simply by counting the number of dots in the picture, then it would be obviously wrong to say 2 was the magnitude of the agreement between the following two people who disagree with each other:



There has to be an agreement and that agreement can be represented by a link. Therefore, it is much more accurate to equate the magnitude of agreement to the number of links in the picture. Note that if there are N people, then it will only require $N - 1$ links to connect them all. Hence the reason for the two “-1” in my formula, which shifted the focus of Morris’ “max-consensus” from “percent of participants” to “percent of agreements”.

To make sure the formula indeed fills the hole left in Table 5, let us revisit a previous situation where we had the following elicitation data:

"id 1": "gesture A",
 "id 2": ["gesture B", "gesture A"]

According to my formula, $P = \{\text{"id 1"}, \text{"id 2"}\}$, therefore $|P| = 2$; $P_{gesture A} = \{\text{"id 1"}, \text{"id 2"}\}$, $P_{gesture B} = \{\text{"id 2"}\}$, thus $\max_{P_i \subseteq P} |P_i| = 2$. This results in an agreement rate of 1 which is justified given that “id 2” does not discriminate between her two proposals. Another example is the “total disagreement” situation:

"id 1": "gesture A",
 "id 2": "gesture B"

Clearly in this case, $|P| = 2$, $\max_{P_i \subseteq P} |P_i| = 1$. The resulting $AR = \frac{1-1}{2-1} = 0$, thus removing the weakness in Morris’ “max-consensus”.

4.4.4 Result.

The alternative agreement formula was used in all the agreement rate calculations on gesture, voice and controller inputs. Of all the tasks performed in the experiment, the mean agreement rate for gestures was 0.6, the one for voice commands was 0.56 and the one for controller was 0.46. Table 6 and the radar chart in Figure 19 give a full picture on the agreement rates with details for each task.

task	type	voice	gesture	controller
aggregate	specification	0.22	0.22	0
compare	specification	0.29	0.29	0.14
disaggregate	specification	0.3	0.5	0.5
filter	specification	0.44	0.22	0.22
sort	specification	0.5	0.38	0.13
highlight	manipulation	0.78	0.67	0.44
right pan	manipulation	0.89	1	0.56
multi-select	manipulation	0.43	0.43	0.43
left pan	manipulation	0.7	1	0.6
rotate	manipulation	0.38	0.88	0.88
single-select	manipulation	0.67	0.8	0.78
zoom in	manipulation	0.75	0.88	0.75
zoom out	manipulation	0.9	0.5	0.5

Table 6 - Input agreement rates for each input mode in all the 13 tasks

Voice, Gesture and Controller Agreement Rates

by task (and type)

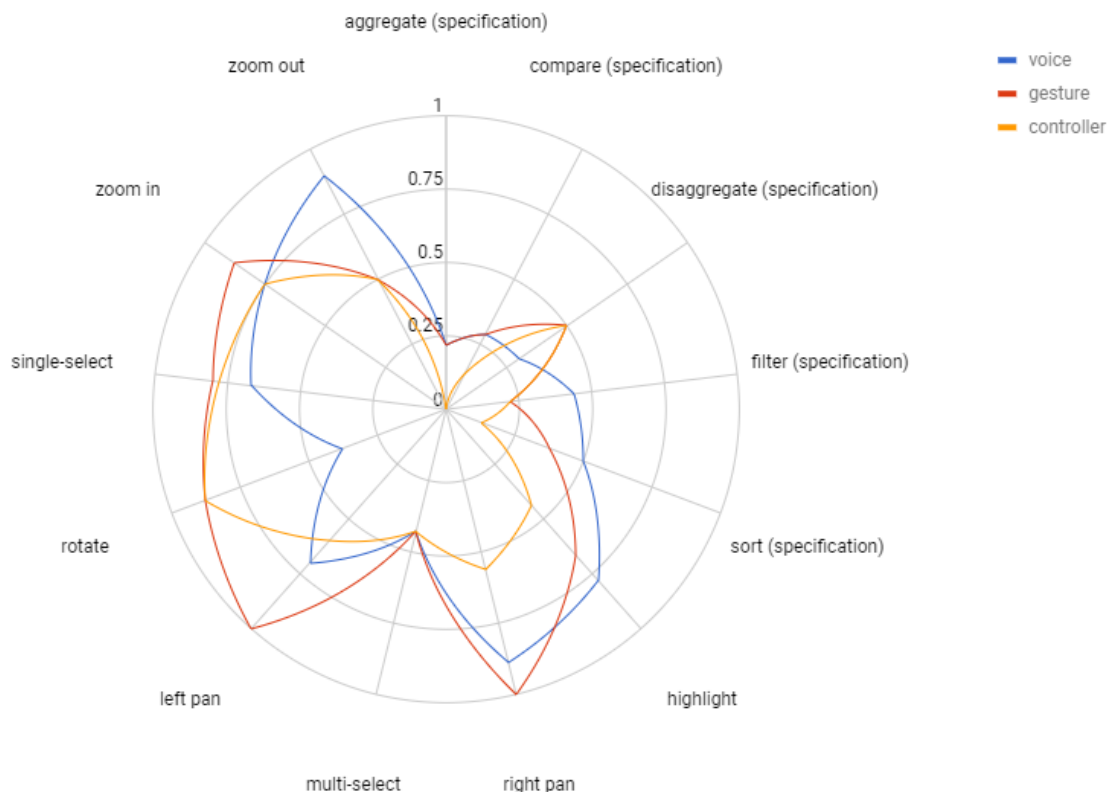


Figure 19 - Another representation of the agreement data from Table 6. The specification tasks were explicitly indicated in brackets. Note the asymmetric nature of this graph due to the large amount of empty space left by those specification tasks, which visually summarizes their overall low agreement rates.

To help answer the second research question of this study, I broke down the result by task type. The resulting mean agreement rate for specification tasks was 0.29 and the one for manipulation tasks was 0.7. To calculate the p-value between them, I used the Mann-Whitney test because 1.) the agreement rate of a specification task has no bearing on that of a manipulation task, they are independent of each other, and 2.) a normal distribution is not assumed for agreement rates. The resulting p-value was 5.952×10^{-6} . Table 7 shows a further breakdown by input modes.

input mode task type	Voice	Gesture	Controller	All
<i>Specification</i>	0.35	0.32	0.2	0.29
<i>Manipulation</i>	0.69	0.77	0.62	0.7

Table 7 - Breakdown of voice, gesture and controller agreement rates by task types

The final table in this chapter shows the most common input (whose count equals to $\max_{P_i \in P} |P_i|$) for each input mode of every task. Note that in some cases there are multiple most

common inputs and those are separated by “|”.

task	type	voice	gesture	controller
aggregate	specification	sum z, group by x	Index finger clicks thumb, hands together	
compare	specification	compare graph A with graph B	Fist, pan, release the fist	both stick turn (stick turn, button x1)x2
disaggregate	specification	scatter disaggregate	Hands apart	stick turn, button x1
filter	specification	keep even x values remove odd x values	(Index finger to screen)x[n] (Fingers together to pick)x[n]	(stick turn, button x1)x[n]
sort	specification	sort all points by z values	Hands folded	stick turn, button x1
highlight	manipulation	X'	Outline	(stick turn, button x1)x[n]
right pan	manipulation	pan right	Index finger traversing	stick turn
multi-select	manipulation	select between [coordinates]	Outline Hand brush	stick turn
left pan	manipulation	pan left	Index finger traversing	stick turn
rotate	manipulation	rotate right [x] degrees	Fingers squeeze, rotate	stick turn
single-select	manipulation	select point with [coordinates]	Index finger click	stick turn, button x1
zoom in	manipulation	zoom in	hands apart	stick turn
zoom out	manipulation	zoom out	Fingers gripping and back	stick turn

Table 8 - Most common inputs for each input mode in every task

Chapter 5 Discussion on Results

Now that we have the results, it is time to revisit the two research questions laid out earlier in Chapter 1 and see what the answers to those are. I will also attempt to explain why the answers are such. Implications of those answer will be explored as well. In addition to that, an effort will be made to highlight the significance of the task types.

5.1 Answers to the Research Questions

5.1.1 Preference question.

My first research question asks,

RQ1: Is there an association between a user's preferred input mode and the type of the task she performs?

My approach to answering this question was to first pick out the most preferred input mode (defined as the one with the largest preference count as seen in Table 3) for each task. Then I focused on p-values from McNemar's test (as shown in Table 4) which relate only to the most preferred input mode of each task. That would give us the following two tables for each type of tasks:

task	type	most preferred input mode	p-value from McNemar's test with Bonferroni correction		
			between voice-gesture pref.	between voice-controller pref.	between controller-gesture pref.
aggregate	specification	voice	0.06	1	-
compare	specification	voice	n/a	n/a	-
disaggregate	specification	controller/gesture	n/a	n/a	n/a
filter	specification	voice	0.09	0.02	-
sort	specification	voice	0.06	0.54	-

Table 9 - p-values from McNemar's test relating only to the preferred input mode for each specification task³

task	type	most preferred input mode	p-value from McNemar's test with Bonferroni correction		
			between voice-gesture pref.	between voice-controller pref.	between controller-gesture pref.
highlight	manipulation	gesture	0.15	-	0.15
right pan	manipulation	gesture	0.12	-	0.39
multi-select	manipulation	gesture	n/a	-	n/a
left pan	manipulation	gesture	<0.01	-	0.06
rotate	manipulation	gesture	n/a	-	n/a
single-select	manipulation	controller	-	0.06	1
zoom in	manipulation	gesture	0.06	-	0.54
zoom out	manipulation	gesture/controller	n/a	n/a	n/a

Table 10 - p-values from McNemar's test relating only to the preferred input mode for each manipulation task

From the two tables, we can see that “voice” was favored for most specification tasks (4 out of 5 in Table 9) whereas “gesture” was the top choice for most manipulation tasks (6 out of 8 in Table 10). However, the apparent associations between “voice” and *specification* tasks as well as the association between “gesture” and *manipulation* tasks are not statistically significant. Most p-values from McNemar’s test in those two tables go above the 0.05 significance threshold.

So the answer to my first research question is, “in most cases, yes, but the association is not statistically significant”. In addition, some outliers such as the specification task

³ “n/a” in both Table 9 and 10 indicates the p-value was not calculated because of a large p-value (> 0.05) resulting from Cochran's Q test.

“disaggregate” (not favoring voice), the manipulation task “single-select” (not favoring gesture) or even “zoom-out” (tied between gesture and controller) may warrant further investigations in the future.

5.1.2 Agreement question.

The answer to my second research question,

RQ2: Are interaction agreement rates for one type of tasks higher than those of the other type?

is a clear “yes”. Table 7 from Chapter 4 shows the participants exhibited a lot more similar interaction behavior when performing manipulation tasks than performing specification ones. The small p-value (< 0.01) between the two mean values of agreement rates (0.70 vs. 0.29) further suggests the validity of the answer.

If we categorize agreement rates into three classes – low, medium and high – and define an agreement rate as “high” if its value is above 0.66 (and “low” if the value is below 0.33), then all the high agreement rates would come from the manipulation tasks and the input encodings associated with them are as follows:

- Gesture
 - Highlight: Outline
 - Right pan: Index finger traversing
 - Left pan: Index finger traversing
 - Rotate: Fingers squeeze, rotate
 - Single-select: Index finger click
 - Zoom in: hands apart

- Voice
 - Highlight: highlight area from [coordinates]
 - Right pan: pan right
 - Left pan: pan left
 - Single-select: select point with [coordinates]
 - Zoom in: zoom in
 - Zoom out: zoom out
- Controller
 - Rotate: stick turn
 - Single-select: stick turn, button x1
 - Zoom in: stick turn

The gestures with high agreement rates share one common trait: they can also be used as body languages to supplement a social conversation between two people. In contrast, low agreement rate gestures, those with an agreement rate below 0.33, such as “(Fingers together to pick)x[n]” for the “sort” task, are not generally employed in a social conversation and may not even be frequently used in a conversation requiring specialized knowledge.

Voice commands with high agreement rates tend to be terse but clear, as shown in the above list. They almost mirrored the task names I gave. This, again, stands in contrast to those with low agreement rates which are either more complex in language structure (e.g., `sum z`, `group by x` for the “aggregate” task) or more ambiguous (e.g., `scatter` for the “disaggregate” task).

The controller inputs with high agreement rates are short and simple, as well. They all involve the turning of a single stick and at most one press on a button. Those with low agreement rates, similar to the low agreement rate voice commands, are more complex and involve many more movements such as `(stick turn, button x1) x[n]` for the “filter” task.

One more thing to note is that none of the manipulation tasks has a low agreement rate (< 0.33) and none of the specification task has a high agreement rate (> 0.66).

5.2 Possible Explanation on Difference in Agreements

In hindsight, it is not too hard to see why inputs for manipulation tasks had a much higher agreement rate than those for specification tasks. In the elicitations, most interaction behaviors for manipulation tasks, whether talking, gesturing or pressing buttons, were mimics of what they would be in an everyday physical environment, which, in the broadest sense, is shared by all people and thus tends to converge behavior. For specification tasks, however, most participants seemed to mimic what they would do in their own minds, which obviously were not physically shared and thus had a much higher chance of producing diverging behavior.

5.3 Implications from Differences in Agreements

The agreement rate data could also serve as a practical, useful guideline to the same UX designer in the field of immersive computing (more applicable if the domain is immersive analytics). Specifically, if the task to design for is manipulation, it might be worth the effort to solicit gestures or voice commands from a small group of potential users, since tasks of that type seem to have higher agreement rates which should make the designer feel more confident

applying the elicited interactions to a broader audience. A similar idea was expressed by Morris as well:

“if the goal is to design a system with a single, highly guessable command... then (a high) max-consensus may be more important” [17]

On the other hand, if the type of a task is specification which suggests a lower agreement rate among users, the designer might skip an elicitation and simply roll out an implementation based on her own idea, which should be further refined based on later feedback.

5.4 Significance of Task Types

This study found a non-statistically significant association between a user’s preferred input mode and the type of task she performed in most cases. In terms of determining the preference for an input mode, the type of a task, at least in the realm of specification and manipulations tasks used in this study, seems to be a factor but may not be a significant one.

A different observation, however, can be made in the agreement rate data. If we lump all the data together, we see agreement level among the three input modes were not that different (0.6, 0.56 and 0.46). This corroborates with aforementioned Morris study in which no statistically significant differences in agreement rates were found between voice and gesture [17]. But Morris did discover that, on a per-task basis, there were some big differences. They went on to suggest that *some* tasks were thus better handled in a particular input mode. My work could be seen as an attempt to make clear what those “some” tasks are. Indeed, after I broke down the agreement rates by manipulation/specification task types, statistically significant differences were revealed.

There may be other ways to differentiate tasks and by no means do I claim the Heer *et al.*'s task taxonomy is the only one, or even a correct one in all cases. However, to further our understanding on all aspects of input modes in immersive computing, I think the characteristics of an underlying task may still offer important clues.

5.5 Comparing Agreement Rates Calculated with My Formula, Max-consensus and Vatavu *et al.*'s Formula

In Chapter 4 I introduced an alternative formula to calculate agreement rates and provided some theoretical arguments for its advantages both to Morris' max-consensus metric as well as Vatavu *et al.*'s formula. In practice, though, how did it really stack up against those two?

For that, I calculated the following agreement rates with both formulas:

task	type	voice	gesture	controller
aggregate	specification	0.3	0.3	0.1
compare	specification	0.38	0.38	0.25
disaggregate	specification	0.36	0.55	0.55
filter	specification	0.5	0.3	0.3
sort	specification	0.56	0.44	0.22
highlight	manipulation	0.8	0.7	0.5
right pan	manipulation	0.9	1	0.6
multi-select	manipulation	0.5	0.5	0.5
left pan	manipulation	0.72	1	0.63
rotate	manipulation	0.44	0.89	0.89
single-select	manipulation	0.7	0.9	0.8
zoom in	manipulation	0.78	0.89	0.77
zoom out	manipulation	0.91	0.55	0.55

Table 11 – Agreement rates according to Morris' max-consensus

task	type	voice	gesture	controller
aggregate	specification	0.02	0	0
compare	specification	0.06	0.03	0.06
disaggregate	specification	0.13	0.13	0.33
filter	specification	0.11	0.13	0.07
sort	specification	0.31	0.11	0.03
highlight	manipulation	0.62	0.49	0.22
right pan	manipulation	0.51	0.82	0.38
multi-select	manipulation	0.08	0.17	0.19
left pan	manipulation	0.62	0.62	0.36
rotate	manipulation	0.17	0.78	0.78
single-select	manipulation	0.33	0.29	0.64
zoom in	manipulation	0.61	0.25	0.58
zoom out	manipulation	0.82	0.22	0.33

Table 12 - Agreement rates according to Vatavu *et al.*'s formula⁴

First of all, none of the agreement rates resulting from “max-consensus” is 0, even though both my formula and Vatavu *et al.*'s formula indicate, at least, the controller input agreement rate for “aggregate” task ought to be 0. This further validates the claim that “max-consensus” does not result in 0.

Second of all, all the agreement rates calculated with Morris' “max-consensus” are slightly inflated compared to the numbers resulting from my formula. This is no surprise given that $\frac{a}{b}$ (max-consensus) is always larger than $\frac{a-1}{b-1}$ (my formula). However, the agreement rates from both calculations share the same ranking order.

The third interesting point is that the numbers from Vatavu *et al.*'s formula is always lower than those from mine, and in some cases, significantly lower. I think there are two reasons for this: one is that in the cases of voice and gesture, my formula simply had more data to work with by having access to at most 3 gestures and voices from each participant, whereas Vatavu *et*

⁴ In order to apply this formula, each participant is supposed to register only one interaction for each input mode. To meet this criteria with the data from my study, only the first gesture/voice from each participant was taken into account.

al.'s formula was restricted to pull data only from the first gesture and voice of each participant. The other reason, in case of controller input mode where both formulas had access to the same amount of data (only 1 controller interaction from each participant), has more to do with the inherent mathematical properties of Vatavu *et al.*'s formula. The formula is much more conservative by squaring a proper fraction, whose value represents the percentage of an interaction among all interactions. Even though Vatavu *et al.*'s formula is more about summing up multiple such squared fractions, there may just not be enough second or third popular interactions in my study to make up for that kind of "loss".

Overall, according to the above analysis based on empirical data, my alternative formula stands on a middle ground in this study: it is a little more conservative than Morris' max-consensus but much more optimistic than Vatavu *et al.*'s formula.

Chapter 6 Limitations, Future Work and Conclusion

This concluding chapter acknowledges current limitations in the study as well as offers suggestions to improve them. It then looks ahead to some possible future work. As the last chapter, it ends fittingly with a summary of the work done throughout the entire thesis.

6.1 Limitations

The first limitation of the study is that 5 of the 21 participants only executed manipulation tasks. Even though the participants were not aware of the type of tasks they performed, given the observation that “voice” was never preferred by those 5 participants, some of them, by the time the 5th or the 6th task was about to be performed, might have felt that “There seems to be only two valid input modes in this study so I am going to stick to one of those two because I do not want to appear as wrong even though for this task I am more inclined toward the 3rd choice.”. This predisposed way of thinking cannot be ruled out. A better way would be simply letting all participants perform a subset of tasks of which the specification vs. manipulation task ratio matches that of the task pool.

The second limitation is the relatively short interaction time the participants had with controllers. The original intentions may still hold (see the “Procedure” section in Chapter 3) but from a study design point of view as well as for the sake of robust statistical results, having a participant use the controller only once per task does leave such a question unanswered: what if the limited exposure to the controller made participants prefer it less? Again, a better way to deal with this would just be to lengthen the elicitation a little to allow for an equal amount of exposure to the controller.

The third limitation is the monologue voice input. Most verbal requests from one person to another result in a dialog between the two because the requestor can rarely express her intentions succinctly in a single utterance, especially for a request with an abstract nature. Yet in my elicitations, I expected the participants to do so to a computer. A better approach could mimic a conversational chat-bot such as the Amazon Alexa to provide a more realistic setting for participants. For example, instead of expecting a participant to say, “sorting by z-axis values”, the researcher should be prepared to follow up her single-word utterance “sort” with the question “by what?” and then expect her to reply with some sorting criteria.

The fourth limitation is the separate use of voice commands and gestures. A more realistic setting is to allow the interwoven usage of voices and gestures (i.e., multi-modal interactions), to issue a request. This way of interaction is a standard feature in human face-to-face communications and it would be ideal not to restrict it in human-to-computer interactions.

Another limitation is the lack of participants with diverse backgrounds and relatively low number of participants. Sixteen of the 21 participants in our study came with a computer science background. While it is certainly creditable to have a computer science professional or student perform visual analytics tasks, it would be more ideal to have participants coming from a wider variety of backgrounds as visual analytics is widely used in other fields as well (natural science and social science). The limited number of participants in our study also means I had to adopt a more cautious tone in narrating my findings.

The last notable limitation of this study is the specific domain of tasks used in the experiment. It confines all the findings in this study within the field of visual analytics. While it is possible to use a list of primitive and domain-agnostic tasks from other studies [12,13] to conduct a similar experiment, those low-level tasks are predominantly manipulation and thus

could lead to a specific result. As pointed out by Nielsen *et al.*, developing a generic, cross-domain approach remains an unsolved problem [11].

6.2 Future work

One obvious follow-up work is to implement a voice and gesture recognition system based on the elicited data and follow the design guideline proposed in Chapter 5. Then a *usability* study could be conducted to validate the findings in this thesis. With today's fast pace of advancement in machine learning, it is possible to assemble a good recognition system with ready-made components such as Google Cloud Speech and Vision APIs.

Another possible work is more theoretical. My alternative agreement rate formula, based on Morris' "max-consensus" [17], inevitably discards information because it only "sees" the most popular input, as indicated by the term $\max_{P_i \subseteq P} |P_i|$. If, for example, the count of the most popular input is 10, then consider the following two cases: in one case, the second most popular input has a count of 9; in the other case, the 2nd most popular input has a count of 0. According to my formula, both cases have the same agreement rate. But the question is: should it be? If not, what can we do next? If there is nothing we could do, does that mean no more elicitation studies where multiple interactions could be proposed? More studies are needed to address those open questions.

6.3 Conclusion

I realized that we could distinguish two types of interaction tasks according to Heer *et al.*'s task taxonomy: specification and manipulation. To find out whether those two types had

anything to do with users' preferred input mode in immersive computing (specifically AR) and whether they had any impact on interaction agreement rates, I elicited gesture, voice and controller interactions from 21 participants wearing a Microsoft HoloLens. After analyzing the video and audio data collected from the elicitation, during which I also developed a formula for calculating agreement rates, I found a non-statistically-significant association between a participant's preferred input mode and the type of tasks she performed in most cases. However, participants did share a lot more similar interaction behavior to execute manipulation tasks than they did to execute specification tasks. An attempt was made to explain why we observed such agreement data and what its implication could be. One AR interaction design guidelines was offered in the process, too. In addition, I believe more understandings on the characteristics of a performed task may lead to further insights on input modes for immersive computing. Finally, with a note on study limitations upon which future work could be based, the thesis concludes.

References

1. Caroline Hummels and Pieter Jan Stappers, Meaningful gestures for human computer interaction: beyond hand postures. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, 1998, pp. 591-596.
2. Marcio C. Cabral, Carlos H. Morimoto, and Marcelo K. Zuffo. On the usability of gesture interfaces in virtual reality environments. In *Proceedings of the 2005 Latin American conference on Human-computer interaction (CLIHC '05)*. ACM, New York, NY, USA, 100-108.
3. Sebastian Pick, Andrew S. Puika and Torsten W. Kuhlen, Comparison of a speech-based and a pie-menu-based interaction metaphor for application control, *2017 IEEE Virtual Reality (VR)*, Los Angeles, CA, 2017, pp. 381-382.
4. Christine Kühnel, Tilo Westermann, Fabian Hemmert, Sven Kratz, Alexander Müller, Sebastian Möller. I'm home: Defining and evaluating a gesture set for smart-home control. In *International Journal of Human-Computer Studies*, Volume 69, Issue 11, 2011, Pages 693-704
5. Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of Oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces (IUI '93)*, Wayne D. Gray, William E. Hefley, and Dianne Murray (Eds.). ACM, New York, NY, USA, 193-200.
6. Roberta C. Ramos Mota, Stephen Cartwright, Ehud Sharlin, Hamidreza Hamdi, Mario Costa Sousa, and Zhangxin Chen. Exploring Immersive Interfaces for Well Placement Optimization in Reservoir Models. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 121-130.

7. Robert Amar, James Eagan, and John Stasko. Low-Level Components of Analytic Activity in Information Visualization. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization (INFOVIS '05)*. IEEE Computer Society, Washington, DC, USA, 15.
8. Jeffrey Heer and Ben Shneiderman. Interactive dynamics for visual analysis. *Commun. ACM* 55, 4 (April 2012), 45-54.
9. Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E. McCullough, and Rashid Ansari. Multimodal human discourse: gesture and speech. *ACM Trans. Comput.-Hum. Interact.* 9, 3 (September 2002), 171-193.
10. Alan Wexelblat. Research Challenges in Gesture: Open Issues and Unsolved Problems. In *Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, Ipke Wachsmuth and Martin Fröhlich (Eds.). Springer Verlag, London, UK, UK, 1-11.
11. Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In *Gesture Based Communication in Human Computer Interaction*. Springer Berlin Heidelberg, 409420.
12. Thammathip Piumsomboon, Adrian Clark, Mark Billingham, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 955-960.
13. 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)*. ACM, New York, NY, USA, 3403-3414.

14. 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 (May 2014), 40-45.
15. Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. Maximizing the guessability of symbolic input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1869-1872.
16. 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)*. ACM, New York, NY, USA, 1325-1334.
17. 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)*. ACM, New York, NY, USA, 95-104.
18. Teddy Seyed, Chris Burns, Mario Costa Sousa, Frank Maurer, and Anthony Tang. Eliciting usable gestures for multi-display environments. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces (ITS '12)*. ACM, New York, NY, USA, 41-50.
19. 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)*. ACM, New York, NY, USA, 1083-1092.
20. Radu-Daniel Vatavu. A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. *J. Ambient Intell. Smart Environ.* 5, 2 (March 2013), 187-211.

21. Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A hand gesture interface device. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*, John M. Carroll and Peter P. Tanner (Eds.). ACM, New York, NY, USA, 189-192
22. Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH '93)*. ACM, New York, NY, USA, 135-142
23. Luis Afonso, Paulo Dias, Carlos Ferreira, Beatriz Sousa Santos. Effect of hand-avatar in a selection task using a tablet as input device in an immersive virtual environment. *3D User Interfaces (3DUI) 2017 IEEE Symposium on*, pp. 247-248, 2017.
24. Michael D. Good, John A. Whiteside, Dennis R. Wixon, and Sandra J. Jones. Building a user-derived interface. *Commun. ACM* 27, 10 (October 1984), 1032-1043.
25. Gourav Modanwal, Kishor Sarawadekar, "A New Dactylology and Interactive System Development for Blind-Computer Interaction", *Human-Machine Systems IEEE Transactions on*, vol. 48, pp. 207-212, 2018.
26. Eduardo Rodrigues, Lucas Silva Figueiredo, Lucas Maggi, Edvar Neto, Layon Tavares Bezerra, João Marcelo Teixeira, Veronica Teichrieb. Mixed Reality TVs: Applying Motion Parallax for Enhanced Viewing and Control Experiences on Consumer TVs. *Virtual and Augmented Reality (SVR) 2017 19th Symposium on*, pp. 319-330, 2017.
27. Leah Findlater, Ben Lee, and Jacob Wobbrock. Beyond QWERTY: augmenting touch screen keyboards with multi-touch gestures for non-alphanumeric input. In *Proceedings*

of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12). ACM, New York, NY, USA, 2679-2682.

28. <https://www.leapmotion.com>
29. <https://www.oculus.com>
30. <https://www.microsoft.com/en-ca/hololens>

Appendix A: User Input Preferences

subject id	standing/sitting pref. when gesturing	task	type	voice	controller	gesture
1	sitting	<i>filter</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>aggregate</i>	<u>specification</u>	best		
		<i>zoom out</i>	<u>manipulation</u>	best		
		<i>zoom in</i>	<u>manipulation</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
2	standing	<i>compare</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>		best	
		<i>aggregate</i>	<u>specification</u>		best	
		<i>highlight</i>	<u>manipulation</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
		<i>zoom in</i>	<u>manipulation</u>			best
3	no pref.	<i>compare</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>	best		
		<i>aggregate</i>	<u>specification</u>	best		
		<i>multi-select</i>	<u>manipulation</u>	best		
		<i>rotate</i>	<u>manipulation</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
4	sitting	<i>sort</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>			best
		<i>compare</i>	<u>specification</u>			best
		<i>zoom out</i>	<u>manipulation</u>	best		
		<i>highlight</i>	<u>manipulation</u>			best
		<i>multi-select</i>	<u>manipulation</u>			best
5	sitting	<i>filter</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
		<i>highlight</i>	<u>manipulation</u>			best
		<i>zoom in</i>	<u>manipulation</u>			best

subject id	standing/sitting pref. when gesturing	task	type	voice	controller	gesture
6	sitting	<i>compare</i>	<u>specification</u>	best		
		<i>filter</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>highlight</i>	<u>manipulation</u>	best		
		<i>left pan</i>	<u>manipulation</u>		best	
		<i>zoom in</i>	<u>manipulation</u>		best	
7	no pref.	<i>aggregate</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>	best		
		<i>filter</i>	<u>specification</u>	best		
		<i>multi-select</i>	<u>manipulation</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
8	sitting	<i>aggregate</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>filter</i>	<u>specification</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
		<i>rotate</i>	<u>manipulation</u>			best
		<i>zoom out</i>	<u>manipulation</u>			best
9	standing	<i>filter</i>	<u>specification</u>	best		
		<i>aggregate</i>	<u>specification</u>		best	
		<i>disaggregate</i>	<u>specification</u>			best
		<i>multi-select</i>	<u>manipulation</u>	best		
		<i>left pan</i>	<u>manipulation</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
10	no pref.	<i>compare</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>	best		
		<i>filter</i>	<u>specification</u>	best		
		<i>zoom out</i>	<u>manipulation</u>	best		
		<i>single-select</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
11	no pref.	<i>aggregate</i>	<u>specification</u>	best		
		<i>filter</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>zoom out</i>	<u>manipulation</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
		<i>highlight</i>	<u>manipulation</u>			best

subject id	standing/sitting pref. when gesturing	task	type	voice	controller	gesture
12	sitting	<i>aggregate</i>	<u>specification</u>	best		
		<i>compare</i>	<u>specification</u>		best	
		<i>disaggregate</i>	<u>specification</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
		<i>rotate</i>	<u>manipulation</u>			best
13	sitting	<i>filter</i>	<u>specification</u>	best		
		<i>sort</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>			best
		<i>multi-select</i>	<u>manipulation</u>		best	
		<i>zoom out</i>	<u>manipulation</u>		best	
		<i>zoom in</i>	<u>manipulation</u>			best
14	standing	<i>aggregate</i>	<u>specification</u>		best	
		<i>compare</i>	<u>specification</u>		best	
		<i>disaggregate</i>	<u>specification</u>		best	
		<i>highlight</i>	<u>manipulation</u>		best	
		<i>rotate</i>	<u>manipulation</u>			best
		<i>single-select</i>	<u>manipulation</u>			best
15	standing	<i>filter</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>		best	
		<i>sort</i>	<u>specification</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
		<i>zoom out</i>	<u>manipulation</u>			best
16	sitting	<i>compare</i>	<u>specification</u>	best		
		<i>aggregate</i>	<u>specification</u>	best		
		<i>disaggregate</i>	<u>specification</u>			best
		<i>highlight</i>	<u>manipulation</u>			best
		<i>multi-select</i>	<u>manipulation</u>			best
		<i>zoom in</i>	<u>manipulation</u>			best
17	sitting	<i>rotate</i>	<u>manipulation</u>		best	
		<i>zoom out</i>	<u>manipulation</u>		best	
		<i>highlight</i>	<u>manipulation</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
		<i>single-select</i>	<u>manipulation</u>			best

subject id	standing/sitting pref. when gesturing	task	type	voice	controller	gesture
18	sitting	<i>right pan</i>	<u>manipulation</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
		<i>highlight</i>	<u>manipulation</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
		<i>rotate</i>	<u>manipulation</u>			best
		<i>zoom in</i>	<u>manipulation</u>			best
19	sitting	<i>rotate</i>	<u>manipulation</u>		best	
		<i>single-select</i>	<u>manipulation</u>		best	
		<i>zoom in</i>	<u>manipulation</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
		<i>zoom out</i>	<u>manipulation</u>		best	
20	no pref.	<i>highlight</i>	<u>manipulation</u>			best
		<i>multi-select</i>	<u>manipulation</u>			best
		<i>rotate</i>	<u>manipulation</u>			best
		<i>zoom out</i>	<u>manipulation</u>			best
		<i>left pan</i>	<u>manipulation</u>			best
		<i>right pan</i>	<u>manipulation</u>			best
21	no pref.	<i>single-select</i>	<u>manipulation</u>		best	
		<i>left pan</i>	<u>manipulation</u>			best
		<i>multi-select</i>	<u>manipulation</u>			best
		<i>rotate</i>	<u>manipulation</u>			best
		<i>zoom in</i>	<u>manipulation</u>			best
		<i>zoom out</i>	<u>manipulation</u>			best

Appendix B: User Input Encodings

Voice Encodings

task	subject	voice1	voice2	voice3
aggregate	1	sum z, group by x		
	2	add all values across y-axis together, make pie chart, distribute it by x	make empty pie chart based on x, add all values of bar, group by x	add all values of same colors, show them in pie chart
	3	sum z, group by x		
	7	get total of each color z values, present them into pie pattern		
	8	sum each x value base them into pie chart	sum by color base them into pie chart	group by sum of x values
	9	for all different colors i see on screen, iterate each color, calculate sum for each color. create pie chart for aggregated sums for each color	create pie chart by aggregating all colors independently	
	11	group by x pie chart	sum each, add up	
	12	sum z values aggregate by x	sum z, group by x	
	14	aggregate into pie chart	sum z values into pie chart group by x	create pie chart using z x values
	16	aggregate into pie chart based on x	aggregate sums	aggregate
compare	2	select A B, compare	select blue red, compare	select everything in this graph, compare
	3	compare with curve B	bring in curve B	load curve B
	4	throw my curve above red curve	drag red curve from right to left	
	6	compare graph A with graph B		
	10	take curve B, compare with curve A	place curve A show it with curve A	compare graph A with graph B
	12	compare curve A with curve B	view curve A curve B	
	14	compare curve A with curve B	view difference between curve A curve B	compare blue curve with other curve
	16	select A B, compare	compare graph A with graph B	

task	subject	voice1	voice2	voice3
disaggregate	3	disaggregate	reset	scatter
	4	scatter	ungroup points	distribute points
	5	disaggregate	scatter	
	7	break down values		
	9	show me individual points	break this part into individual points	
	10	show original data points	show data of values	show original data points on graph
	12	disaggregate		
	13	disperse	separate	
	14	break apart	scatter	
	15	disaggregate	show me individual points	
	16	disperse	break apart	
filter	1	filter by even X values	remove odd x values	keep even x values
	5	filter by even X values	remove odd x values	keep even x values
	6	filter by even X values	filter by odd x values	(filter out color)x3
	7	keep even x values	remove odd x values	(filter out color)x3
	8	remove odd x values	keep even x values	
	9	for each odd x bars, make it disappear	starting from 1 bar, alternatively remove each bar	(filter out color)x3
	10	filter out X1, X3, X5	filter out X1, X3, X5	(filter out color)x3
	11	filter out X1, X3, X5		
	13	filter by odd x values	keep even x values	remove odd x values
	15	filter by odd x values		
highlight	2	highlight area from [coordinates]	draw rectangle select everything inside it, [coordinates]	
	4	highlight area from [coordinates]		
	5	highlight area from [coordinates]		
	6	highlight area from [coordinates]		
	11	show rectangle less or more by [x] amount	multi-select less or more by [x] amount	

task	subject	voice1	voice2	voice3
highlight	14	highlight area from [coordinates]	select all points between [coordinates]	get all points between [coordinates]
	16	highlight area from [coordinates]		
	17	highlight area from [coordinates]		
	18	highlight area from [coordinates]		
	20	highlight leftmost [x] percent of points, except 1st point		
left pan	6	pan left	scroll left	move left
	7	move purple section to left	change position of purple blue	put purple section in middle
	8	pan left	rotate left	show y-axis
	9	move data points toward center so that left point align with points [coordinates]	move data points to left until i say stop	
	12	pan left	move left	shift left
	15	pan left	move left	
	17	pan left	move left	
	18	pan left	move to right	see more on right
	19	pan left	move left	
	20	pan left	move image to right	slide to right
	21	move it bit along axis to left		
multi-select	3	select between [coordinates]	select everything below [coordinates]	
	4	select between [coordinates]	select base	
	7	select points on ground	select points with 0 z value	
	9	select points with [coordinates]	move height of 4000 select all points underneath that	
	13	select between [coordinates]	group x plain	select points with 0 z value
	16	select everything below [coordinates]	select between [coordinates]	
	20	select bottom [x] percent of points		
	21	select all points under [coordinates]		
right pan	1	pan right	move blue points to left side, move yellow points to left side	
	2	move all points toward value axis	move whole graph towards left	move whole graph to center

task	subject	voice1	voice2	voice3
right pan	3	pan right	over left	move left
	10	pan right	rotate	view left
	12	pan right	move right	shift right
	15	pan right	move right	
	17	pan right	move right	
	18	move to left	pan right	view more on left
	19	pan right	move right	
	20	pan right	move to left	slide left
rotate	3	rotate	over	
	8	rotate right [x] degrees		
	12	rotate then stop	rotate graph	frontview
	14	rotate right [x] degrees	spin left	
	17	rotate [x] degree counter-clockwise	rotate graph to left little bit	
	18	rotate right [x] degrees	(turn right)xN	
	19	rotate [x] percent to left	rotate chart to look at me	
	20	rotate right	rotate counter clockwise	
	21	rotate right [x] degrees	rotate right	rotate so that Y axis is more visible
single-select	5	select point with [coordinates]		
	7	select point with relative	select point with [coordinates]	
	9	select point with [coordinates]	At [x] percent left to (0,0) point, select point	descend at [x] percentage stop
	10	select point with [coordinates]		
	11	select point with [coordinates]	select middle, then select left or right to point	
	14	select point with [coordinates]		
	17	select point with [coordinates]		
	18			
	19	select [x]-th point to left	pick [x]-th point to left	
	21			
sort	1	sort all points by z values		
	2	sort all points by z values	convert to 2d without y, sort by value in asc order	take out y, sort by/based on values

task	subject	voice1	voice2	voice3
sort	4	sort all points in vertical line, starting from 0	sort all points, pick line, starting from 0	arrange all points, pick line, starting from 0
	5	sort according z values, in asc order	arrange all points, from small to large, wrt z values	order z values from small to large
	6	sort all points by z values	sort points, arrange by z values	
	8	sort all points by z values	plot on xy graph, by z values	
	11	sort z		
	13	sort z	group	
	15	sort all points by z values		
zoom in	1	zoom in	specify coordinate, zoom in	
	2	zoom in [x] percent	make graph twice its original size	expand
	5	zoom in	expand	detail
	6	zoom in	expand	
	13	zoom in	move closer	enlarge
	16	zoom in	inspect	move closer
	18	zoom in	go in	
	19	zoom in	enlarge around this area	take me closer
	21	zoom in [x] percent	make it [x] inch by [x] inch	
zoom out	1	zoom out	go back to original state	
	4	zoom out	go far	
	8	zoom out	pan out	
	10	zoom out	enlarge	make bigger
	11	zoom out	expand	unfocus
	13	zoom out	smaller	
	15	zoom out		
	17	zoom out	make it small	
	19	zoom out	take me further away	
	20	zoom out	pan out	move back
	21	make it little bit smaller		

[coordinates] = arbitrary x, y and z values; [x] = arbitrary numeric values

Gesture Encodings

task	subject	Gesture1	gesture2	gesture3
aggregate	1	Fingers close together, picking x[n], fingers loose, hands spin	(fingers together to move)x[n], both hands draw half circle	
	2	The index finger clicks, the index finger draws a circle, and the palm of the hand touches the screen.	Fingers close together, move straight up, index finger click, index finger circle, slap palm	Slap to screen, straight forward, index finger click, index finger circle, palm to screen
	3	Index finger clicks thumb, fists	(press down, hand shift)x[n]	Index finger clicks thumb, hands together
	7	(Forefinger clicks on thumb)x[n], hands at the same time draw half park	Fingers are knife-shaped, various movements, both hands simultaneously draw half garden	
	8	(Forefinger clicks on the thumb)x[n], fingers together, picking x[n]		
	9	((index finger click)x[m], index finger tabs thumb)x[n]		
	11	One-hand fist, one-handed circle		
	12	Fist close	Hands folded up and down, pinch rotating	Hands folded horizontally and pinched
	14	Index finger clicks thumb, hands together	Index finger clicks thumb, fists	
	16	One fist, hands folded	One-handed circle, hands folded	Index finger clicks thumb, hands together
compare	2	index finger double click, palms up and alternately move up and down	The thumb of the index finger opens, moves along the curve, and the palms of the hands move upwards and alternately move up and down	index finger double click, hands together
	3	Fist, pan, release the fist		
	6	Fingers together to pick and move		
	4	Forefinger thumb open, move along the curve	Fingers together to pick and move	

task	subject	Gesture1	gesture2	gesture3
compare	10	Fist, pan, release the fist		
	12	Hands close together		
	14	One-handed index finger clicks thumb, another hand flips	Index finger double click, hands together	
	16	One hand fist	Index finger double click	Fist, pan, release the fist
disaggregate	3	Hands apart	High-five	Close up
	7	boxing	Boxing down	Left and right uppercuts
	10	Fight back	Hands apart	
	4	Index finger click	Hands are parallel to each other, with both hands index finger at the same time, with both hands simultaneously index finger click	High-five
	5	High-five	Index finger click	Forefinger thumb is horizontally separated
	9	High-five		
	15	Five fingers apart	Hands apart	
	16	Hand waving back and forth	Five fingers apart	Hands apart
	12	Hands apart		
	13	Hands apart	Five fingers apart	Waving once
	14	Hands apart	Hands off	
	filter	1	(Index finger to screen)x[n]	
5		(Index finger to screen)x[n]		
6		Vertical separation of hands		
7		(Fingers together to pick)x[n]		
8		(Fingers together to pick)x[n]		
9		(Index finger clicks x[m], fists)x[n]		
10		(Fingers together to pick)x[n]	((Index finger clicks)x[m], fists)x[n]	
11		One hand on hold, the other hand ((index finger click)x[m], fist)x[n]		

task	subject	Gesture1	gesture2	gesture3
filter	13	One hand down		
	15	(Index finger to screen)x[n]		
highlight	2	Outline	Smear rectangle	Hands with your index finger thumb squeezed and dragged down
	4	Outline	Hands cut together and fingers grip back	
	5	Outline	Diagonal	
	6	Hands apart	Forefinger thumb separated	
	11	Diagonal	Forefinger thumb separate and diagonal	Hands apart
	16	Diagonal	Select one corner with one finger and choose one corner with the other	Click on 4 corners
	14	Outline	Diagonal	Z word
	18	Outline	Forefinger thumb separated	Smear rectangle
	20	Outline	Smear rectangle	Click on 4 corners
	17	Outline	Hands forefinger into a square	
right pan	7	Index finger traversing	both hands traverse	
	9	Index finger traversing		
	8	Index finger traversing		
	6	Index finger traversing		
	12	Index finger traversing		
	15	Index finger traversing		
	19	both hands traverse	Index finger traversing	Waving to the palm of other hand
	20	Index finger traversing	Index finger double-click the other side	
	18	Index finger traversing	both hands traverse	
	17	Index finger traversing		
multi-select	21	Index finger traversing		
	3	Fingers close together	Forefinger thumb tight, hands separated	Hands are knife-shaped while separating
	4	Outline	(Index finger click)x[n]	Outline x2

task	subject	Gesture1	gesture2	gesture3
multi-select	7	Outline	Hand brush	Hands down, hands separated
	9	(Index finger clickf)x[n]	Two index fingers	
	13	Outline	Palm lift	Hand brush
	16	Diagonal stroke	Hand brush	Forefinger thumb tight, hands separated
	20	Outline	Forefinger thumb separation, translation	
	21	Hand brush	Two-handed fists, translation, rings	
left pan	1	Index finger traversing	Fingers squeeze, rotate	
	2	Index finger traversing		
	3	Index finger traversing		
	10	Fingers squeeze, rotate	Index finger traversing	
	12	Index finger traversing		
	15	Index finger traversing		
	17	Index finger traversing	Hands traverse	
	18	Index finger traversing		
	19	Hands traverse	Index finger traversing	Waving to the palm of your hand
	20	Index finger traversing	Index finger double-click the other side	
rotate	3	Fingers squeeze, rotate	Forefinger thumb squeeze, pan	Hands moving horizontally at the same time
	8	Fingers squeeze, rotate		
	12	Fingers squeeze, rotate		
	14	Fingers squeeze, rotate		
	17	Fingers squeeze, rotate	Palm horizontal move	
	20	Fingers squeeze, rotate	Palm horizontal move	
	21	Fingers squeeze, rotate		
	18	Fingers squeeze, rotate		

task	subject	Gesture1	gesture2	gesture3
rotate	19	Two-handed fists, rotating	Two-handed rotation	
single-select	5	Index finger click		
	7	Index finger click		
	9	Index finger click	Forefinger thumb tight	
	10	Hands open, index finger click	Forefinger thumb tight	
	11	One hand stopped in the air, the other hand moved	Index finger click	Forefinger thumb tight
	14	Index finger click	Forefinger thumb tight	
	18	Forefinger thumb tight	Index finger click	
	19	Forefinger thumb tight	Index finger click	
	21	Forefinger thumb tight	Index finger click	
	17	Index finger click	Forefinger thumb tight	
sort	1	Hands folded	Hands with index finger thumb squeezed	
	2	Fingers click, hand wave to one side, another hand	Fingers click, palm forward, another hand	Fingers click, hand wave to one side, knife up another hand
	4	Hands folded	Hands with index finger thumb squeezed	
	5	One hand life up		
	6	One hand diagonal up	Hands folded	
	8			
	13	Hands folded	One hand pressure	Hands down
	11	One hand diagonal up	Hands tamper	
	15	moves Each point	Finger stroke Z	
zoom in	1	Forefinger thumb open	hands apart	
	2	Forefinger thumb open	hands apart	Palm open
	5	Palm open	Index finger double click	Forefinger thumb open
	6	hands apart	Hands up and down	
	13	hands apart	Hands and fingers separated at the same time	
	16	hands apart	Palm open	Fingers squeezed with your index finger, hands lifted
	19	hands apart	Palm open	Hand parabolic

task	subject	Gesture1	gesture2	gesture3
zoom in	21	Fingers squeezed with your index finger, hands lifted	Forefinger thumb open	hands apart
	18	Forefinger thumb open	Index finger double click	hands apart
zoom out	1	Forefinger thumb tight	Compression with both hands	
	4	Compression with both hands	Fingers gripping and back	Forefinger thumb tight
	10	Hands separate at the same time	Fingers gripping and back	
	8	Forefinger thumb tight		
	11	Forefinger thumb tight	Hands separate at the same time	Fingers gripping and back
	15	Compression with both hands	Double click	
	13	Hands separate at the same time	Fingers open	Fingers gripping and back
	17	Compression with both hands	Compression with both hands	
	19	Compression with both hands	Fingers gripping and back	
	20	Fingers open	Hands separate at the same time	Fingers gripping and back
	21	Hands separate at the same time	Fingers open	

$x[n]$, $x[m]$ = repeating the preceding action arbitrary number of times

Controller Encodings

task	subject	movement
aggregate	1	button x1, stick turn $x[n]$, button x1
	2	button x1, stick turn, button x1
	3	button x4
	7	stick turn, button x2, d pad, stick turn
	8	button x1, button $x[n]$, stick turn
	9	(d pad, stick click x1, stick turn, stick click x1) $x[n]$, sbutton x1, stick turn, stick click x1
	11	d pan, stick turn, button x1, stick turn, button x1,
	12	menu button, button x1, d pad, button x1, menu button, d pad, button x1
	14	stick turn, x button x1, y button x1
	16	menu button, stick turn, button x1, menu button, stick turn, button x1

task	subject	movement	
compare	2	stick turn, button x1,	
	3	sbutton x1	
	4	(stick turn, button x1)x2	
	6	both stick turn	
	10	both stick turn	
	12	button x1, d pad, button x1	
	14	(hold stick, sbutton x1)x2	
	16	(stick turn, button x1)x2	
disaggregate	3	button x1	
	4	stick turn, button x1	
	5	stick turn, button x1	
	7	stick turn, button x1	
	9	stick turn, button x1	
	10	stick turn, button x1	
	12	stick turn, button x1, stick turn, button x1	
	13	button x1	
	14	sbuttonx2	
	15	button x1	
	16	stick turn, button x1	
filter	1	stick turn, button x1	
	5	(stick turn, button x1)x[n]	
	6	sbutton	
	7	(stick turn, button x1)x[n]	
	8	(stick turn, button x1)x[n]	
	9	(d pad, sbutton, stick turn, stick click)x[n]	
	10	(d pad, stick turn)x[n]	
	11	d pad, (stick turn, button x1)x[n]	
	13	button x1, (d pad, button x1)x[n]	
	15	button x1, stick turn	
highlight	2	d pad, stick turn	
	4	stick turn, button x1	
	5	(stick turn, button x1)x[n]	
	6	(stick turn, button x1)x[n]	
	11	hold sbutton, stick turn, release sbutton	
	14	(stick turn, button x1)x[n]	
	16	(stick turn, button x1)x[n]	
	17	d pad x4, button x1	

task	subject	movement
highlight	18	stick turn, sbutton x1
	20	(stick turn, button x1)x[n]
left pan	6	stick turn
	7	stick turn
	8	stick turn
	9	stick turn
	12	stick turn
	15	stick turn
	17	d pad x[n]
	18	sbutton x[n]
	19	sbutton
	20	button x1, stick turn
	21	stick turn
multi-select	3	button x1, d pad
	4	stick turn
	7	stick turn
	9	stick turn
	13	stick turn
	16	(stick turn, button x1)x[n]
	20	button x1, stick turn
21	(stick turn, button x1)x[n]	
right pan	1	stick turn
	2	stick turn
	3	stick turn
	10	stick turn
	12	stick turn
	15	stick turn
	17	d pad x[n]
	18	d pad x[n]
	19	sbutton
	20	button x1, stick turn
rotate	3	stick turn
	8	stick turn
	12	stick turn
	14	stick turn
	17	stick turn
	18	stick turn
	19	sbutton
	20	stick turn
21	stick turn	

task	subject	movement
single-select	5	stick turn, button x1
	7	stick turn, button x1
	9	stick turn, button x1
	10	stick turn, button x1
	11	stick turn, button x1
	14	stick turn, button x1
	17	d pad, button x1
	18	d pad, button x1
	19	stick turn, button x1
	21	stick turn, button x1
sort	1	stick turn, button x1
	2	stick turn, button x1, button x1, stick turn, button x1
	4	(stick turn, button x1)x[n]
	5	stick turn, stick turn, button x1
	6	move both stick
	8	stick turn, button x1
	11	d pad, button x1
	13	button x2
	15	button x1, d pad, button x1
	zoom in	1
2		hold button, stick turn
5		stick turn
6		sbutton
13		stick turn
16		stick turn
18		stick turn
19		stick turn
21		stick turn
zoom out	1	stick turn
	4	stick turn
	8	
	10	stick turn
	11	sbutton
	13	stick turn
	15	sbutton
	17	d pad
	19	stick turn
	20	stick turn
21	sbutton	

x[n] = repeating the preceding action arbitrary number of times; sbutton = shoulder button