Enhancing Collaboration and Productivity for Virtual and Augmented Reality

by

Zhenyi He

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science
New York University
January, 2021

Ken Perlin

To my parents, with affection.

Acknowledgements

I would not be here without the many excellent advisors, collaborators, and friends throughout my research career, who have helped shape me into the person I am.

To begin with, I want to thank my advisor, Professor Ken Perlin, and unofficial advisors who advised my projects very deeply. Ken Perlin is not only a brilliant computer science pioneer, but also a very inspiring and considerate mentor. Every time I traveled with him and listened to his talk, I felt very motivated and positive about the technology in the future.

I am very lucky to work with quite a few resourceful professors and researchers. I met Professor Christof Lutteroth on CHI because of mutual friends. Later, Christof advised me on project TapGazer for almost one year. He is so encouraging and patient. He helped me polish the paper sentence by sentence. He comforted me when I got rejected. He is so practical which influenced me a lot.

I got introduced to work with Professor Xiaojuan Ma a while ago. I started to understand user study, human-computer interaction, data analysis in an official way because of her. Until now, I bothered Xiaojuan for naive user study questions once a while.

I want to especially thank Ruofei Du, a Google research scientist, who I met on CHI too. We have so many research interests in common and successfully work on a paper together during covid-19.

To my labmates and visitors, Aaron Gaudette, Connor Defanti, Fengyuan Zhu, Amber Hu, Ben Ahlbrand, and Gabriel Nunes. Big thanks for being there working with me, working until

sunrise with me, proofreading the document for me, and brainstorming with me.

I am grateful to my parents, who understand me and encourage me when I am studying abroad individually.

Abstract

Immersive environments such as Virtual Reality (VR) and Augmented Reality (AR) are now receiving more and more attention. Although VR and AR have largely been used for individual entertainment experiences, they also possess huge potential as a platform for the support of collaboration and productivity. My thesis work is concerned with enabling VR/AR to be flexibly adapted for collaborative and productive uses. I approach this scope from several facets: a new haptic user interface based on actuated robots to bridge virtual and physical world, a reconfigurable framework for both co-located and geographically dispersed multi-user communication, and a text entry system in which users type by tapping their fingers, without needing to look at their hands or be aware of their hand positions. Further, I extend these ideas to a daily video conferencing experience that requires minimal hardware.

Contents

List of Figures

List of Tables

1 Introduction

With the development of head-mounted displays(HMDs) in the past decade, Virtual Reality (VR) and Augmented Reality (AR) has reached a greater level of popularity and familiarity than ever before. However, the absence of effective haptic feedback will strongly detract from the suspension of disbelief needed to bridge the virtual and physical worlds. Since participants do not directly observe these robotic proxies, the multiple mappings between physical robots and virtual proxies are worth researching for various setups, such as individual use cases and distributed collaboration.

While in the middle of researching collaborative use cases, I notice that the way how users communicate with each other is not always efficient. For example, sometimes, we need to sketch on whiteboards for better clarity, but someone's line of sight may be blocked. I enlarge the scope that could bring benefits to collaboration for general use. More specifically, users in immersive environments have full control of the contents and even other participants. The way how we manipulate the content and other representations have the potential to improve communication for various use.

Nevertheless, controllers are standard input solution to immersive environments, even for typing or writing. As a computing platform, an immersive environment has its advantage to enhance the user's daily and working manner. Typing or writing through a cumbersome controller is not ideal. Thus, provide a text entry solution that allows typing anywhere efficiently is crucial.

In this thesis, I will focus on three aspects that need to be addressed to achieve this vision: 1) providing haptic feedback, 2) enhancing collaboration and communication via immersive environments, and 3) text entry solutions.

1.1 Providing Haptic Feedback

Haptic feedback can be a powerful component of immersion and can be used to improve user experience in VR to great effect. Followed by concept "Robotic Graphics" [McNeely 1993] described that "robots simulating the feel of an object and graphics displays simulating its appearance", which has opened new opportunities for enhancing experiences in Virtual Reality. There is much current research on haptic feedback from passive objects [Benko et al. 2016; Kajita et al. 2016; Otsuki et al. 2010; Sugihara et al. 2011], human actuated systems[Cheng et al. 2015], and actuated or wheeled robots[Araujo et al. 2016; Cheng et al. 2015; Iwata et al. 2005; Le Goc et al. 2016; Pedersen and Hornbæk 2011a], but a number of limitations remain.

First, passive objects and human actuated systems support only static haptic feedback, which is insufficient when interacting with a dynamic environment. Also, those approaches do not support dynamic mapping between physical props and their virtual proxies, so in many cases, they might require a large number of actuated objects with a consequent increase in system complexity.

Other approaches have been developed for distributed workspaces [Brave et al. 1998; Rosenfeld et al. 2004; Riedenklau et al. 2012; Ishii and Ullmer 1997; Ishii 2008]. Some offer different views for users [Follmer et al. 2013; Gauglitz et al. 2014; Leithinger et al. 2014, 2015; Sra 2016; Sra et al. 2016], while others share the same environment without haptic feedback [Orts-Escolano et al. 2016; Sra and Schmandt 2015].

I focus on variable mappings between virtual objects in VR worlds and their physical robotic proxies, multiple modes of physical manipulation (direct manipulation, remote synchronization, and the illusion of telekinesis), and a solution for distributed collaboration based on wheeled robots.

1.2 Communication through Immersive Collaboration

As we know communication is happening every day and everywhere, locally and remotely. However, it is not always very effective. For example, sometimes, we need to sketch on whiteboards for better clarity, but someone's line of sight may be blocked.

Despite recent advances in collaborative work in virtual reality (VR), exchanging ideas between users is mostly achieved through direct media such as audio [Hsu et al. 2020] and video [Otsuka 2016], or indirect media such as gestures [Tversky et al. 2003] and scene editing [Huo et al. 2018]. Sketching, one of the most natural and fun ways to express ourselves, has rarely been explored in collaborative VR. Additionally, it is an open question of what is the best layout and interaction mode for creative collaboration.

Hence, I focus on researching and designing a collaborative framework that is flexible for the various collaborative use case, like presentation and brainstorming, and address the challenge of enhancing communication efficiency.

1.3 Text Entry

As VR and AR become more and more popular, they have the potential to become the future computing platform. In addition to entertainment use, which was greatly explored in VR and AR, productivity is also an important topic for serious use. Thus, we choose text entry as a starting point. Text entry is one of the most frequent, important, and demanding tasks in personal computing. Because efficient text entry methods are crucial to productivity, an enormous amount of research has been conducted on methods that improve their usability [Dudley et al. 2019].

Text entry solutions were researched with the development of devices. New text entry methods have been proposed [Dunlop et al. 2012] as new types of electronic devices such as smartphones have become available.

In addition to devices, many alternative keyboard layouts [Rick 2010; Zhai et al. 2002] have been proposed to optimize typing speed or energy, e.g. ATOMIK [MacKenzie and Zhang 1999] and Quasi-Qwerty [Bi et al. 2010]. For devices where a physical QWERTY keyboard is not available, many specialized text entry solutions have been proposed, e.g. for touch screens [Li et al. 2011], mobile phones [Dunlop et al. 2012], and other handheld devices [Castellucci et al. 2019]. Speech-to-text is also a widely explored option with the potential to be faster than typing [Ruan et al. 2018]; however, it has limited accuracy and not always suitable, e.g. when the environment is noisy, other people are talking, or the content is of a sensitive or personal nature.

I focus on investigating a new text entry method designed to address text entry in HMD environments. Thus, users can perform text entry simply by tapping their fingers, without needing to look at their hands or be aware of finger position.

1.4 Document Structure

In the following chapters, I will introduce my research into researching each of these areas, showing that it is valuable to provide haptic feedback, enhance collaborative VR, and text entry solutions that facilitate and benefit from Virtual Reality and Augmented Reality. chapter 2 provides an overview of the literature and prior work domains that intersect with my thesis work. chapter 3 describes my initial explorations multiuser network framework: Holojam. chapter 4 describes an end-to-end solution for providing haptic feedback in VR coped with various scenarios. PhyShare provides tabletop solutions and room-scale solutions as well. In chapter 5, I start to research how to enhance communication in general through VR or AR. After exploring Chalktalk, I decide to build a multiuser system that provides rich interaction and manipulation on top of creative software like Chalktalk. Two scenarios, teaching, and creative sensemaking are evaluated. chapter 6 addresses how to improve productivity with head-mounted displays(HMDs) via a novel text entry solution. Typing with HMDs has its specific challenge which users have difficulty seeing their body in real and the surrounding environments. With TapGazer, users are able to type without knowing the positions of the hands and input devices like keyboards. Unexpectedly, I turn my attention to building a videoconferencing system with an ordinary setup to enhance gaze awareness for remote group discussion. The system, LookAtChat, detailed in chapter 7, is a web-based videoconferencing system that only requires a webcam. Exploratory design space is proposed to discuss the variations of providing gaze awareness as well as manipulating the spatial information among the participants. Finally, in chapter 8, I conclude by summarizing key contributions and defining future avenues of exploration into this space.

2 Background

My research work intersects with many diverse areas of human-computer interaction research. First, I will review work that discusses providing haptic feedback for individual or collaboration use, specifically prior robotic graphic literature, and work that shares haptic feedback remotely. Next, I cover work that focuses on enhancing communication for collaborative use. Lastly, I review prior arts on improving productivity in a head-mounted display environment.

2.1 Physical Objects in Virtual Reality

Physical objects have different forms of interactions and effects on Virtual Reality to provide haptic feedback to users. The mechanisms to apply physical objects into immersive experiences can be generalized into the following categories.

To begin with, *Passive Object* is the topic where the real objects are mapped to the virtual environment. SkyAnchor is an example of attaching virtual images to physical objects that can be moved by the user[Kajita et al. 2016]. Liao et al. implemented a telemanipulation system that simulates force feedback when controlling a remote robot manipulator[Liao et al. 2000]. Moreover, *Robotic Shape Display*[McNeely 1993] refers to the scenario in which actuated robots provide feedback when users come into contact with a virtual desktop. This scenario requires the robot to be ready to move at any time to meet the user's touch. For example, TurkDeck[Cheng et al. 2015] simulates a wide range of physical objects and haptic effects by using human actuators.

NormalTouch and TextureTouch[Benko et al. 2016] offer haptic shape rendering in VR, using very limited space to simulate feedback of different surfaces of various objects. Snake Charmer[Araujo et al. 2016] offers a robotic arm that moves based on its user's position and provides corresponding haptic feedback of different virtual proxies with different textures and temperatures. Last but not least, *Roboxels* (robotic volume elements) is the area where robots can dynamically configure themselves into a desired shape and size. For example, CirculaFloor[Iwata et al. 2005] provides the illusion of a floor of infinite extent. It uses movable floor elements, taking advantage of the inability of a VR user to see the actual movement of these floor elements.

While this work addresses the interactions of physical objects in different settings and cases of haptic feedback, it does not address the strategies of physical setups in a more general and dynamic environment where the number of users and robots may vary. Understanding how robots in different quantities can be mapped to a VR environment allows the abovementioned work to be enhanced in a more scalable manner.

2.1.1 Tabletop TUI and Swarm UI

TUIs allows users to manipulate physical objects that either embody virtual data or act as handles for virtual data[Richter et al. 2007]. TUIs can assume several different forms, including passive sensors[Otsuki et al. 2010; Sugihara et al. 2011; Choi and Follmer 2016] and actuated pieces[Araujo et al. 2016]. They incorporate physical objects moving on a flat surface as input[Everitt et al. 2003] or are used to simulate autonomous physical objects[Brave et al. 1998; Leithinger et al. 2014; Pedersen and Hornbæk 2011b; Riedenklau et al. 2012; Rosenfeld et al. 2004]. There are examples of variations. Tangible Bots[Pedersen and Hornbæk 2011a] use wheeled robots as input objects on a tabletop rear-projection display. When the user directly manipulates them, they in turn react with movement according to what the user has done. Tangible Bits[Ishii 2008; Ishii and Ullmer 1997] is a general framework proposed by Hiroshi Ishii to bridge the gap between cyberspace and the physical environment by incorporating active tangible elements into the interface. Zooids[Le Goc et al. 2016] provides a large number of actuated robots that behave as both input and output. Since Zooids merge the characters of the controller and haptic display, users can perform manipulations more freely.

2.2 Presence of physical objects in VR

Prior research has shown that, by careful calibration, VR designers can deliberately incorporate certain activities in the physical world, such as real walking [Iwata et al. 2005; Sun et al. 2016], drawing [Otsuki et al. 2010; Sugihara et al. 2011] and touching the physical entity of a virtual object [Stone 2001] as a part of social and environmental presence, to enhance the sense of immersion and presence in virtuality. Sun et al. discussed the mapping between the physical world and virtual world [Sun et al. 2016]. Considering different room sizes, wall shapes, and surrounding objects in the virtual and real worlds, it attempted to warp the virtual world appearance into real-world geometry, for example how a physical table became a virtual wall in users' VR experience.

However, prior work mostly focused on (1) presenting the physical objects including passive objects [Liao et al. 2000] and actuated systems like NormalTouch [Benko et al. 2016], PhyShare [He et al. 2017], SnakeCharmer [Araujo et al. 2016] and TurkDeck [Cheng et al. 2017, 2015, 2014] in VR for haptic feedback or direct manipulation; Or (2) presenting human beings as another player in VR for social interaction and collaboration research[Hoyer et al. 2004], which has a different identity from our work. NormalTouch [Benko et al. 2016] provided direct manipulation through physical objects. PhyShare [He et al. 2017] created a different mapping between the virtual proxy and physical robots and controlled the robots to provide instant haptic feedback to indicate the existence of physical objects. It visualized the object by a similar representation. SnakeCharmer [Araujo et al. 2016] offered different textures to mimic different objects so that users felt differently when touching them. All objects are rendered as cubes which the same as the physical object itself. TurkDeck [Cheng et al. 2017, 2015, 2014] is a multiuser experience, however users play as main-actor in their own VR experience and play as part of the environment in other's scenarios. The design of the experiment avoids users to have interactions with each other.

2.3 Distributed Collaboration

There are distributed systems that offer different views for different users [Follmer et al. 2013; Gauglitz et al. 2014; Leithinger et al. 2014, 2015; Sra 2016; Sra et al. 2016]. An example of applications [Gauglitz et al. 2014] provides spatial annotations for local users as guidance. Other work shares the same environment without haptic feedback, including Holoportation [Orts-Escolano et al. 2016] and MetaSpace [Sra and Schmandt 2015], in which all interactions are established as belonging either to local physical objects or remote virtual objects. InForm [Follmer et al. 2013; Leithinger et al. 2014, 2015] introduces an approach to physical telepresence that includes capturing and remotely rendering the shape of objects in shared workspaces. Local actuated movables behave both as input by manipulation and output by shape-changing. However, all of these works do not address haptic feedback for remote users. Moreover, PSyBench[Brave et al. 1998] first suggested the possibility of distributing TUI. The approach was to virtually share the same objects when users are remote using Synchronized Distributed Physical Objects. A similar idea called Planar Manipulator Display[Rosenfeld et al. 2004] was developed to interact with movable physical objects. A furniture arrangement task was implemented for this bidirectional interaction. Also, Tangible Active Objects[Riedenklau et al. 2012] extends the idea by adding audio feedback. This research was based around the constraint of What You Can See is What You Can Feel[Yokokohji et al. 1996]. There is work mentioning different mapping possibilities in distributed collaboration[Reilly et al. 2011, 2010; Richter et al. 2007]. TwinSpace[Reilly et al. 2011, 2010] extends the idea of the "office of the future" [Raskar et al. 1998] by presenting a framework for collaborative cross-reality. It supports multiple mixed reality clients that can have different configurations.

2.4 Immersive Collaboration

The collaboration was pointed out as one of the important topics in a recent survey [Kim et al. 2018]. During the past, co-located and remote immersive collaboration systems have been developed. Multi-user entertaining experiences is one trend. For example, Popovici and Vatavu [Popovici and Vatavu 2019] examined users' preferences for AR television scenarios. Increasing engagement for single user [He et al. 2018] and for sharing museum experience was widely discussed [Franz et al. 2019]. Haptic feedback is investigated for remote collaboration [He et al. 2017]. Remote guidance is popular for AR and VR collaboration, such as exploring visual communication cues [Kim et al. 2019], creating virtual replicas of local objects for remote experts [Elvezio et al. 2017], updating remote objects based on local users' actions [Thanyadit et al. 2018], and providing multiple view sharing techniques [Lee et al. 2020].

Developing telepresence experiences for bridging the gap between the physical and virtual worlds plays a vital role in remote collaboration. Teo *et al.* [Teo et al. 2019] explored mixing 360 video and 3D reconstruction for remote collaboration. MetaSpace [Sra and Schmandt 2015] performed full-body tracking. Young and Cook [Young et al. 2019] provided a hand overlay on a panoramic reconstruction. Holoportation [Orts-Escolano et al. 2016] demonstrated real-time 3D reconstructions of an entire space with a comprehensive setup of eight cameras and gigabyte-level bandwidth. Beck *et al.* [Beck et al. 2013] implemented immersive group-to-group telepresence, which allowed distributed groups of users to meet in a shared virtual 3D world through two coupled projection-based setups. Similarly, Pejsa *et al.* [Pejsa et al. 2016] presented Room2Room, a telepresence system that leverages projected AR to enable life-size, co-present interaction between two remote participants. SharedSphere [Lee et al. 2018] was implemented to investigate how Mixed Reality (MR) live panorama reconstruction affects the remote collaborative experience with non-verbal cues.

In addition, collaborative tools, such as editing [Bergig et al. 2009], manipulation [Grandi et al. 2018], modeling [Weichel et al. 2014], and information analysis [Cavallo et al. 2019; Butscher et al. 2018], were proposed for productive work in immersive environments. Hsu *et al.* [Hsu et al. 2020] developed an architecture design discussion system that allows members to visualize, discuss, and modify the architectural models. Members communicate via voice, object manipulations, and mid-air sketching as well as on-surface sketching. Object manipulation and navigation were under research for decades. T(ether) [Lakatos et al. 2014] was a spatially-aware display system for co-located collaborative manipulation and animation of objects. T(ether) attached trackable markers on the pads so participants with gloves can interact with the objects through gestures. Kunert *et al.* [Kunert et al. 2019] designed an application to support object manipulation tasks and scene navigation. Oda *et al.* [Oda et al. 2015] developed a distributed system for remote assistance. Geollery [Du et al. 2019; Du and Varshney 2016] focused on social experiences by creating an interactive MR social media platform. Mahmood *et al.* [Mahmood et al. 2019] presented a remote collaborative visualization system by providing co-presence, information sharing, and collaborative analysis functions to discuss complex problems like environmental pollution.

For collaborative purposes like social networking and telepresence, engagement, and a sense of being there are the most important qualities. In those scenarios, communication performance is not the focus. While for collaborative purposes, such as productive work, games, assistance, and object manipulation, which require complicated and specific operations and information exchange, communication performance becomes more important. In CollaboVR, our goal is to build a reconfigurable framework to fit different purposes of creative collaboration, including side-by-side whiteboarding, face-to-face demonstration, and lectures with a presentation.

2.5 Communication in Immersive Environments

We researched the trends of communication in immersive environments. Asymmetrical communication was under discussion for scenarios that not all the participants use the same device [Grandi et al. 2019]. ShareVR [Gugenheimer et al. 2017] enabled the communication between an HMD user and a non-HMD user. Through floor projection, the non-HMD user can interact with the HMD user and become part of the VR experience. Mutual human actuation [Cheng et al. 2017] ran pairs of users at the same time and had them provide human actuation to each other. Communication between the pair was through the shared interactive props. Avatar representation plays an important role [Layng et al. 2020]. Mini-Me [Piumsomboon et al. 2018] was an adaptive avatar representing the remote user's gaze direction and body gestures. Chow *et al.* [Chow et al. 2019] identified several challenges for time-distributed collaborators in asynchronous VR collaboration. Maintaining workspace awareness is one challenge.

Interacting with digital content in a shared space also triggers a line of in-depth research. Kiyokawa *et al.* [Kiyokawa et al. 2002] have researched the communication behavior for two participants in collaborative AR. They found that placing the task space between participants led to the most active behaviors through an icon design task. "Three's Company" [Tang et al. 2010] explored three-way collaboration over a shared visual workspace. They illustrated the utility of multiple configurations of users around a distributed workspace. TwinSpace [Reilly et al. 2010] supported deep interconnectivity and flexible mappings between virtual and physical spaces. Sra *et al.* [Sra et al. 2018a] proposed "Your Place and Mine" to explore three ways of mapping two differently sized physical spaces to shared virtual spaces and to understand how social presence, togetherness, and movement are influenced. Irlitti *et al.* [Irlitti et al. 2019] discussed how to design and provide spatial cues to support spatial awareness in immersive environments for remote collaboration. Likewise, Volmer *et al.* [Volmer et al. 2018] provided projector-based predictive cues to improve performance and to reduce the mental effort for procedural tasks. Tan *et al.*

[Tan et al. 2010] built a face-to-face presentation system for remote audiences. Lukosch *et al.* [Lukosch et al. 2015] pointed out that face-to-face collaboration increased social presence and allowed remote collaborators to interact naturally. Tele-Board [Gumienny et al. 2011] described a groupware system focused on creative working modes using a traditional whiteboard and sticky notes in digital form for distributed users. Benko *et al.* [Benko et al. 2014] proposed a unique spatial AR system that enables two users to interact in a face-to-face setup. Thanyadit *et al.* [Thanyadit et al. 2019] presented ObserVAR to discuss gaze awareness and visual clutter for VR classroom.

2.6 Text Entry

The QWERTY keyboard is the most widely used text entry method in English speaking countries. Originally designed around the hardware limitations of early mechanical typewriters, QWERTY is not optimal for modern keyboard technology. As a result, many alternative keyboard layouts [Rick 2010; Zhai et al. 2002] have been proposed to optimize typing speed or energy, e.g. ATOMIK [MacKenzie and Zhang 1999] and Quasi-Qwerty [Bi et al. 2010]. For devices where a physical QWERTY keyboard is not available, many specialized text entry solutions have been proposed, e.g. for touch screens [Li et al. 2011; Kim et al. 2013; Shi et al. 2018], mobile phones [Dunlop et al. 2012; Zhu et al. 2018], and handheld devices [Castellucci et al. 2019]. Moreover, using a finger [Blumrosen et al. 2020; Parizi et al. 2019] or pen [Kristensson and Zhai 2004] for handwritten text input has been considered, although this is slow compared to typing. Speech-to-text is also a widely explored option with the potential to be faster than typing [Ruan et al. 2018]; however, it has limited accuracy and is not always suitable, e.g. when the environment is noisy, other people are talking, or the content is of a sensitive or personal nature.

3 | Initial Exploration: Holojam

Holojam is a framework for wireless shared-space virtual reality. It enables people to be positionally tracked and physically interacted with each other in a shared environment. Drawing, dancing, theater experiences were experimented on on the top of such spirit.

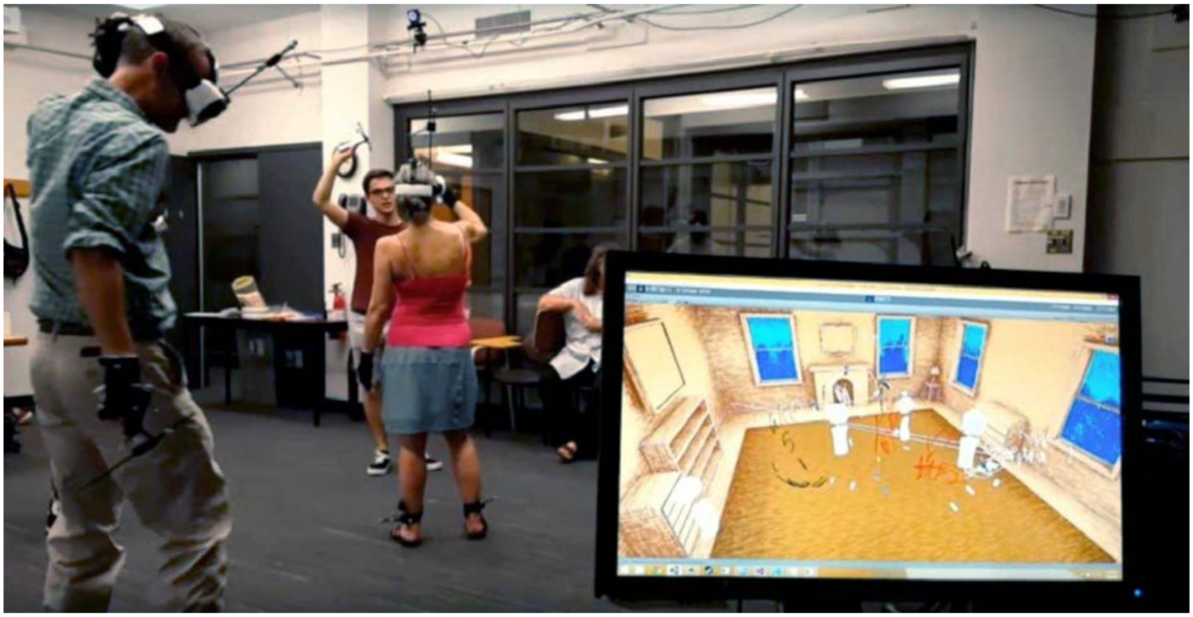


Figure 3.1: Physical setup and virtual rendering of Holojam framework.

Holojam[Perlin 2016] is an untethered virtual reality headset system that enables people to have a shared-space VR experience. Holojam has been demonstrated in the SIGGRAPH 2015 performance session. Briefly: each participant needs to put on a lightweight wireless motion-tracked GearVR headset as well as strap-on glove markers, tracked by OptiTrack. These devices allow them to see everyone else as an avatar, walk around the physical world, and interact with real physical objects.

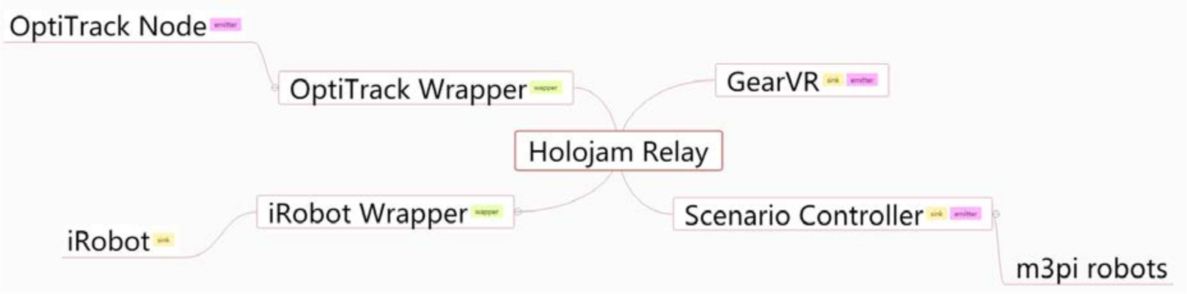


Figure 3.2: Holojam network architecture.

The novel "Rainbows End" by Vernor Vinge [Vinge 2006] described a very specific vision of this future: One day we will all be able to wear a pair of cyber-enhanced contact lenses that will allow us to see whatever we wish to see. In this version of the future, we will continue to walk around the world on our own two feet, socially co-inhabiting the physical world with each other. But that physical world will be visually transformed in ways that are limited only by the imagination. But how will we get to that reality?

There are at least two ways that this can happen: Either by broadcasting VR images to headsets or by having the needed graphics run within headsets themselves. The first of these two approaches, represented today by such initiatives as [MASON 2016], will not scale to large numbers of people sharing the same physical space. The second approach is indeed scalable, as it avoids the impractically high bandwidth requirements of the first approach.

But in order for this latter scenario to become a practical reality, networks will need to be optimized for shared VR. That will require high reliability, low latency wireless communication.

A number of recent systems have aimed to approximate this vision, allowing people to freely walk around in a physical space with their own two feet while inhabiting a shared virtual reality without the restriction of being physically tethered to a computer. In such "walk-around VR" systems, each participant maintains a local computation device which generates a viewpoint for that participant, and also wirelessly receives up-to-date state information of other participants, thereby allowing each participant to view the other participants as avatars.

Some existing systems approximate this scenario by having participants wear backpacks connected to a head-mounted display (HMD) containing hardware support both for graphic computation and communication via a wireless network. For example, RealVirtuality[Interactive 2015], VRcade[VRcade 2015] and The Void [VOID 2016], require each participant to wear a backpack, to contain the PC that computes each participant's view and communicates wirelessly with a server.

We have improved on this by developing a custom wireless protocol that enables low latency and high-reliability wireless communication on mobile devices. This capability has enabled us to develop "Holojam", a backpack-free approach, which betters supports empirical research into future scenarios of embodied social interaction since it requires participants to wear only lightweight headsets and unintrusive tracking markers. We overcame obstacles such as real-time wireless networking on commodity hardware, graphical processing on low-powered phones, calibration, and sensor fusion to provide a novel experience whereby users inhabit space together and collaboratively construct virtual 3D artwork.

We have achieved this by combining sensor fusion with the use of ordinary commodity level Android phones running their original operating system, as both the means for computing the view of each participant and the means for receiving the real-time position and orientation updates of all of the other participants in the scene.

4 Sharing Physical Interaction in Virtual Reality

4.1 Introduction

Because of the rich visual experience it provides, VR is suitable to create simulations and illusions [He et al. 2018]. Meanwhile, as we all know, the way we understand the world is mediated by our five senses: touch, taste, sound, smell, and sight [1]. Although visual feedback in VR strongly increases our immersive experience, the lack of other sensational components creates a cognitive gap between the real world and virtual world [He et al. 2018]. Some previous work has been done to explorer to what extent, different senses can strengthen the belief of the real world. Among

[1] https://en.wikipedia.org/wiki/Sense

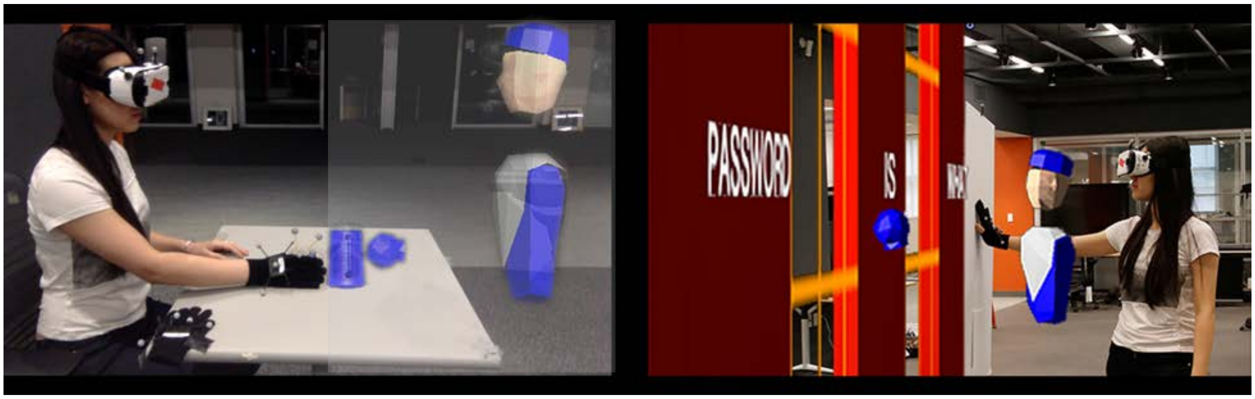


Figure 4.1: Sharing physical interaction when clinking the Mugs and Pushing the Wall

the five senses other than sight, touch is the most common sense to be used in regular life [Benko et al. 2016]. Therefore, nowadays haptic feedback is of paramount importance to create new bridges to narrow the gap between the virtual world and the physical world.

Creating haptic feedback, however, is by no means a trivial task. There are various researches for VR to provide multimodal experience [Riedenklau et al. 2012]. Researches have been focusing on passive objects [Benko et al. 2016; Kajita et al. 2016; Otsuki et al. 2010], actuated robots [Le Goc et al. 2016; McNeely 1993], or other special options like human beings [Cheng et al. 2014, 2015, 2017]. While these works explore various niche settings of virtual environment and activities, there is a more general yet challenging design problem: The scale of the virtual world can grow beyond space and interactions. By space, it means that users can experience a universe scale of the environment inside a head-mounted display, while interactions mean that a single virtual object can be shared by a great number of users at the same time but in different locations. When designing an immersive environment with haptic feedback, the first question is how should we represent virtual items with physical objects? Answering this question leads to the utilization of haptic feedback, immersive experience, and physical resource allocation at the same time.

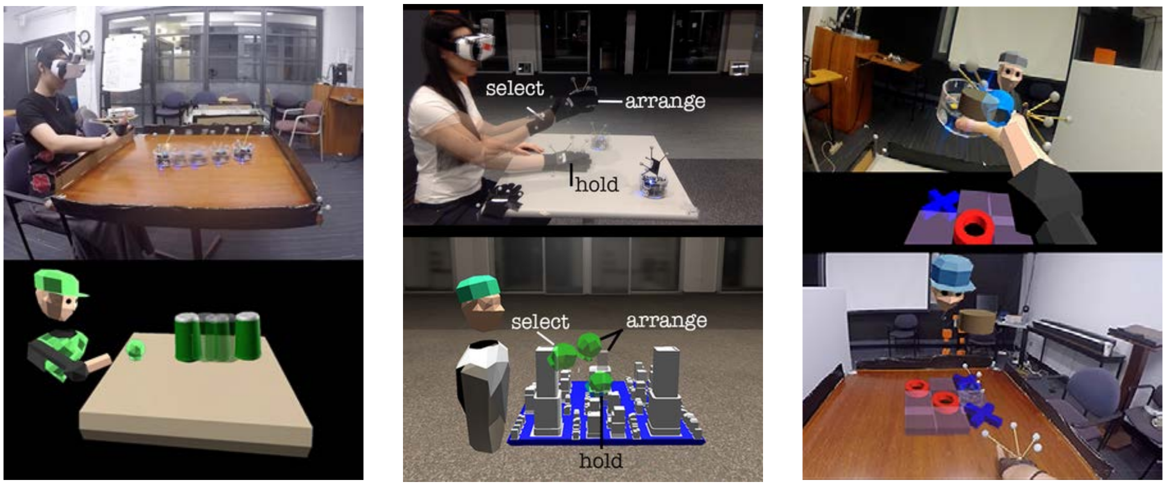
Indeed, analyzing the utilization of physical and virtual objects can be treated as a mapping process. The task is to simply find an ideal way to map a varying amount of virtual objects to a fixed amount of physical devices or vice versa. To be specific, if the objects in the experience are all realized, the cost will become large. Yet considering the capability of creating an illusion by VR, each virtual representation with haptic feedback design does not have to be one-to-one mapped to a physical proxy. For example, if we want to create a maze in a large playground, we could use moving walls to establish a closed environment. On the other hand, if we want to bridge people all over the world to share the same physical interaction in real-time, we could represent a shared virtual object with multiple devices. Such mappings between physical and virtual create synergy between physical and virtual experiences.

As a result, we propose PhyShare, a novel haptic user interface based on actuated robots, that provides different mapping mechanisms to cope with various situations dealing with the above-mentioned utilization challenges. We implement various immersive experience activities, ranging from small scale tabletop games to large scale moving activities and from local interactions to remote collaborations, to demonstrate different design considerations, implementations, and user experience that are a result of our mapping principles.

4.2 Design

The main challenge of providing physical feedback of a virtual environment is the method used to map virtual objects to their physical counterparts (or vice versa). In non-VR TUI systems, there is generally a direct mapping between what users see and what they feel. In tangible VR systems, there is more room to experiment with the relationship between real and virtual environments. Conventionally, virtual objects have no physical representations and can only be controlled indirectly or through the use of a standard input device (such as a game controller). When users do not directly observe the physical input devices, we can represent N virtual objects with 1 device, or to use N devices to represent 1 virtual object. However, the relationship between devices and virtual objects is not linear. Using one or multiple devices and virtual objects will fall into different usage scenarios.

We propose a mapping mechanism to address the challenges. The relationships between physical and virtual representation can either be: 1) *one-to-one*, which maps the interactions between one physical object and virtual object; 2) *one-to-many*, which bridges one physical object to various virtual objects, and; 3)*many-to-one*, which addresses the situations when multiple physicals represent one virtual component. These mappings are defined based on the activity setup, as well as the concerns related to the immersive experience created, which we are going to discuss below.



(a) telekinesis **(b)** city builder **(c)** Tic-Tac-Toe

Figure 4.2: PhyShare Use Cases. (a) shows an one-user scenario that user can manipulate the object via gesture control. (b) illustrates how user arranges a miniature through multiple robot proxies. (c) captures a remote Tic-Tac-Toe game.

4.2.1 One-to-One Mapping

This is the standard and straightforward mapping which is applied in most of the scenarios of haptic feedback. When users interact with a virtual object in the scene, they simultaneously interact with a physical proxy at the same location. We illustrate such mapping in the 'telekinesis' use case (figure 4.2(a)), the m3pi robot represents the virtual mug. Grasping the moving bot representing a virtual mug shares similarity of VR scenarios where users only need to interact with one object. Such mapping leads to a fundamental question of sharing physical interactions in VR: To what extent is such mapping *necessary* to immersive experience? In general, one-to-one mapping has the goal to provoke the designer's awareness of the necessity of the whole setup.

4.2.2 One-to-Many Mapping

Various constraints, including cost and space, limit the capability of maintaining a physical counterpart for everyone of a large number of virtual objects. When fewer proxies are available than virtual objects, one of the total available proxies could represent a given virtual object as required.

For example, a user with a disordered desk in VR may want to reorganize all of their virtual items. In each instance of the user reaching to interact with a virtual item, the single (or nearest, if there are multiple) proxies would relocate to the position of the target item, standing by for pickup or additional interactions. The virtual experience is seamless, while in the physical world a small number of proxies move into position as needed to maintain that seamless illusion. We illustrate this scenario in the "city builder" use case (figure 4.2(b)). A proxy is fitted behind-the-scenes to the virtual building which is visually nearest to a given user's hand. The movement and position of the robots are completely invisible to the user. Also, to scale the scenario up, we present the "escape the room" use case, where the iRobot carries a physical wall that moves with the user to provide haptic feedback. The virtual wall that the user perceives in VR is much longer than the touchable wall section that represents it in the physical world (figure 4.1).

The main concern that comes to the implementation and mapping is *feasibility*. How well can the robots behave to make such mapping happen? The goal of our interface is to cope with latency and physical limitations in the immersive environment.

Retargeting. We extend the idea of Synchronized Distributed Physical Objects[Brave et al. 1998] and adapt it to a VR environment based on a retargeting system [Azmandian et al. 2016]. The virtual object usually does not necessarily represent the physical location of the robot, except while the user is handling the object. In such a case, both remote users and the operating user can observe the real time movement of the physical object. At other times, the remote robot follows the euclidean path to the closest virtual object to the user.

4.2.3 Many-to-One mapping

When multiple proxies represent one virtual object, we define the mapping as "many-to-one". This is useful for remote-space applications: A virtual object could exist in the shared environment, which could then be manipulated by multiple users via their local proxies. The "clink the mugs" use case simulates the effect of two users striking their mugs together while in separate physical locations (figure 4.1). In such a scenario, multiple physical objects are synchronized to bring stimulation of collision from the same virtual object. Also, the “Tic-Tac-Toe” use case (figure 4.2(c)) applies such mapping as a remote game. In one user’s turn, he or she interacts with the virtual environment by moving the chess, while the opponent can observe the whole movement of the same piece in another location.

Performing physical interactions in different locations but in the same virtual environment produces a concern for *distance*. The objective for such mapping is to narrow the perceived distance between users through physical feedback, which is useful when the activities in VR are indeed close in real life. To address such user experience issues, our focus on implementation lies in creating an artificial latency.

Manual Delay. We add a “manual” delay to virtual objects to improve the user experience when synchronizing remote actions since the robotic proxy takes time to move into position. Given the speed of the robot and the size of the tabletop workspace, we need to provide some delays to give the robot sufficient time to catch up with each user’s actions. Movements by the local user of their proxy object are visually delayed in the remote collaborator’s view. This helps to smooth the haptic operations considerably.

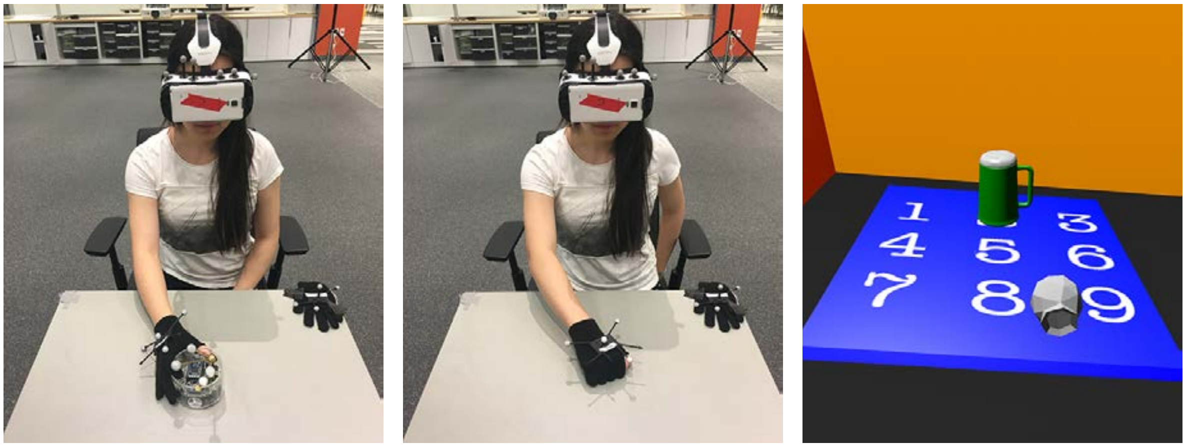
4.2.4 Manipulation

The manipulation of a physical object is crucial to the necessity issue of one-to-one mapping. Therefore, our system supports several methods of interacting with virtual proxies via physical objects. Our system should support a gesture recognition technique, enabling users to command the position of a target using hand motions. Utilizing a simple configuration of tracked markers that reports each user’s wrist position and orientation, users can push a nearby(proximate) object across the table via a pushing gesture, pull a distant object towards them with a pulling gesture, or motion in another direction to slide the object towards another part of the table. The system should also support a direct one-to-one manipulation, which allows a grab and goes mechanism of physical objects.

4.3 Experiment

While *one-to-many* mapping mainly deals with the technical feasibility challenges of physical barriers and limitation, *one-to-one* mapping and *many-to-one* mapping address the user's necessity of physical objects and the user's perceived distance to other users respectively. As a result, we conducted a user study to address these issues. The goals of our study are (1) *Necessity.* How do users perceive one-to-one physical interaction and what kind of physical manipulation do they prefer? (2) *Awareness of latency.* How much latency does the user perceive when interacting remotely with another user? (3) *Awareness of remote location of the other participant.* Does the user feel that their opponent is sitting at the same physical table, or does it feel as though their opponent is remotely located?

Sixteen participants took part in our study. Thirteen of them were male and three were female. Ages ranged from twenty-two to fifty-two, and the median age was twenty-six. All participants were right-handed. Seventy-five percent had experienced VR before the study.



(a) physical Tic-Tac-Toe with controller to make the move

(b) pure Tic-Tac-Toe with only hand to make the move

(c) table split by 9 areas for moving the mug

Figure 4.3: Experiment Sketch

4.3.1 Experiment Design and Procedure

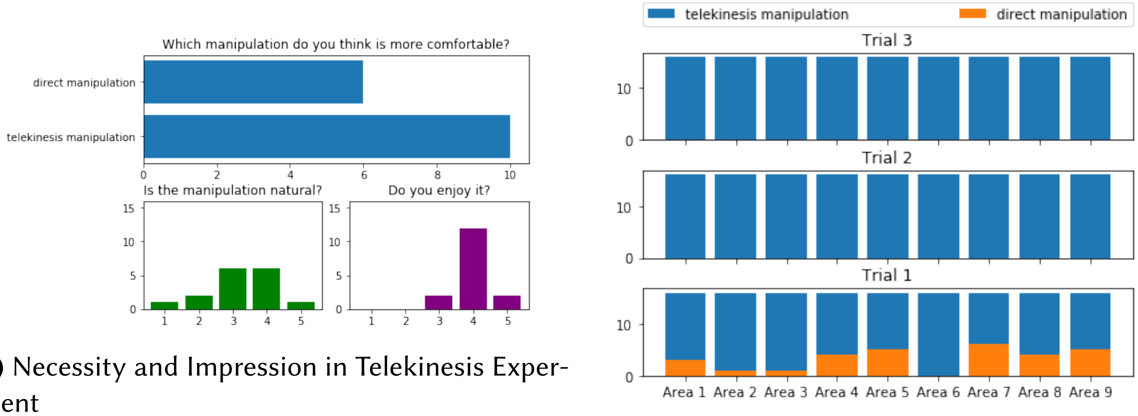
We designed two tests for our user study. In the first experiment, "Telekinesis" (figure 4.3(c)), we split the virtual table into nine parts, sequentially placing a target in each one of the subsections. Users will repeat doing this three times. They could either use gestures to control the robot or directly grasp it. Its purpose was to measure users' ability to learn our gesture system and to use it to command a target at various locations, such that they can provide a preference and habit for physical interactions. Participants were first given up to two minutes to learn the gestures, then we observed their command choice (including non-gestural physical interaction) for each target position.

The second was a recreation of the classic board game, *Tic-Tac-Toe*, in VR. We utilized a "controller" object, allowing players to select a tile for each game step. When one player is holding the controller and considering their next move, the other player can observe this movement in their view as well. We included a version of *Tic-Tac-Toe* that was purely virtual (no haptics) for comparison (figure 4.3(a) and 4.3(b)). This allows us to examine whether they would be distracted by latency and whether they would experience a closer distance from their counterparts.

To begin with, we first explained to participants the purpose of the study. Before each task, the interaction techniques involved were described. Then when participants were doing the tasks, we record the statistics for each experiment. In a questionnaire, the participants rated their interaction using four questions from QUIS[Chin et al. 1988] to give impressions for the whole activity on a Likert scale.

4.4 Results

Figure 4.4(b) shows the results of the preference of manipulating physical objects in the telekinesis experiment. Out of 16 manipulations in the first trial, further areas (i.e area 1,2,3) were being



(a) Necessity and Impression in Telekinesis Experiment

(b) Preference of Telekinesis Manipulation

Figure 4.4: Telekinesis Experiment

directly manipulated by 3,1 and 1 times respectively; middle areas (i.e. area 4,5,6) were 4,5 and 0 times, and nearest areas (i.e. area 7,8,9) were 6,4 and 5 times. For the rest of the trials, participants finished all with hand gestures. Figure 4.4(a) shows that slightly more participants prefer gestures over direct manipulations in the telekinesis experiment. Although participants only slightly agree that the gesture system is natural (M = 3.25, SD = 0.97), they almost agree that this is enjoyable (M = 4.0 , SD = 0.5)

We believe this result has two implications. First, *the necessity of one-to-one physical mapping depends on the convenience of interaction.* When participants are close to the physical props, they are more eager to opt for haptic feedback in the VR environment. This is illustrated in the experiment that participants had the willingness to move the virtual objects in a more certain way when they are closer. Second, *the necessity of one-to-one physical mapping also depends on the certainty of interaction.* When the participants became familiarized with the physical set up after the first trial, they all began to use some more VR oriented interactions. When the perceived similarity between virtual and physical objects became closer, the necessity of physical mappings might decrease.

Figure 4.5 reveals that participants experienced a strong presence of remote opponents (14 out of 16) in the immersive environment. Also, they showed a little awareness of latency during

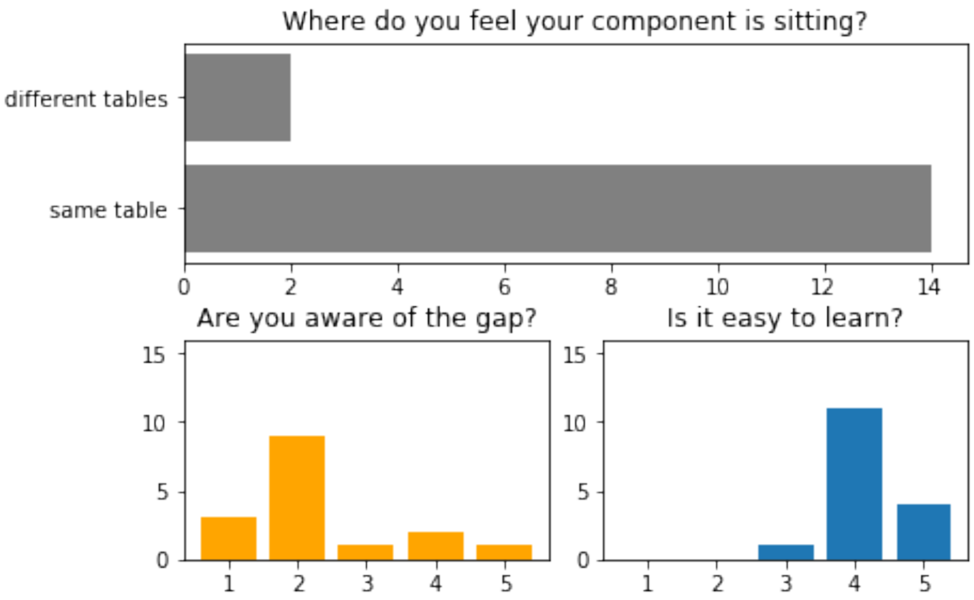


Figure 4.5: Acknowledgement of placement and Latency in Tik-Tac-Toe Experiment

the process ($M = 2.31$, $SD = 1.10$) with a general agreement that the interaction is easily to understand ($M = 4.19$, $SD = 0.527$).

Such a result encourages the improvement of perceived distance between users when multiple physical objects are mapped to represent the same virtual object. Synchronizing distant interactions provides a more realistic feeling of human interaction to users. We believe that such a mapping mechanism has a great potential to introduce social activities in the immersive environment.

4.5 Discussion and Conclusion

All of the mappings we proposed were shown effective under certain technical and user circumstances. *One-to-one mapping* is shown necessary when users need to perform an unfamiliar task or experience a new immersive environment. *One-to-many mapping* can provide a decent economical resource allocation under certain computation and physical criteria. *Many-to-one mapping* provides the possibility to perform more social activities in the VR environment due to the effectiveness of narrowing the perceived distance between users' locations. To demonstrate that our interface is useful to the VR environment, we created various scenarios including clicking the mugs, playing Tik-Tac-Toe, city builders and telekinesis, and walking through artificial walls. These scenarios are realized with our combination of hardware and software setup that can be assembled by market available products.

Undoubtedly, our work has several limitations. First of all, we evaluate only three physical mappings, omitting alternate versions. For example, physical mapping can also be applied to a fixed environment, or even incur even human as one of the proxies. Our mapping only emphasizes the mapping between robots and virtual objects without considering other stakeholders.

Apart from that, we did not explore any moving algorithms for robots related to our proposed mappings. While our work mainly focuses on the scenario, design, consideration, and outcomes between the relationships of physical objects and their virtual representation, we believe that moving algorithms will play a more vital role when the mapping is non-linear. For example, Sun et. al. proposed a nonlinear moving algorithm of users when a floor plan is provided[Sun et al. 2016]. A moving algorithm that effectively distorts the virtual environment for minimizing physical movement will be our next direction of work.

Overall in the past two years, I have been demonstrating the usage scenarios in various venues, including public lectures and leading HCI conferences. The feedback is generally positive. The main attraction for our system is the efficiency of robots to represent different virtual identities. One of the visitors mentioned that "I thought everything I interacted with was manual until I realized there's just a single robot doing all the stuff! " When asking about the suggestions for improvement, some of the audiences mentioned that they would like to see more examples of physical mappings on a larger scale. Having a collaborative system of *one-to-many* mappings could have an overwhelming experience of VR and a broader range of applications. In the future, we would like to develop an algebra among mappings to examine the scalability of our approaches.

We proposed a new approach for interaction in virtual reality via robotic haptic proxies, specifically targeted towards collaborative experiences, both remote and local. We presented several prototypes utilizing our three mapping definitions, demonstrating that robotic proxies can be temporarily assigned to represent different virtual objects, that our system can allow remotely located users to have the experience of touching on the same virtual object, and that users can alternately use gestures to command objects without touching them. Our preliminary experiments returned positive results. In the future, we plan to undertake a more comprehensive study focusing on deeper application interactions and expanded hardware capability.

5 | Enabling Reconfigurable and Creative Collaboration in Virtual Reality

5.1 Preliminary Experiment: Extending Chalktalk

There exist limitations and potential for PhyShare including more mapping mechanisms, more flexible robot designs, better robot movement control, better scalability, or drones. While designing and experimenting with the PhyShare Tic-Tac-Toe, I realized the importance of collaboration – connecting people via technology. Like "office of the future", I next focused on how to enhance communication via immersive environments.

Chalktalk [Perlin et al. 2018a,b] is an open-source presentation and visualization tool in which the user's drawings are recognized as animated and interactive "sketches", which the user controls via mouse gestures. Sketches help users demonstrate and experiment with complex ideas during a live presentation without needing to create and structure all content ahead of time. Chalktalk allows a presenter to create and interact with animated digital sketches in order to demonstrate ideas and concepts in the context of a live presentation or conversation. For each raw sketch, Chalktalk first matches its strokes with the most similar one in a library of 150 glyphs. We designed our own glyph for experiment use. Based on the recognized glyph pattern, it further converts the raw sketch into digital objects that the user can manipulate. We illustrate examples of real-time conversion from raw sketch to animated objects in Figure 5.1.

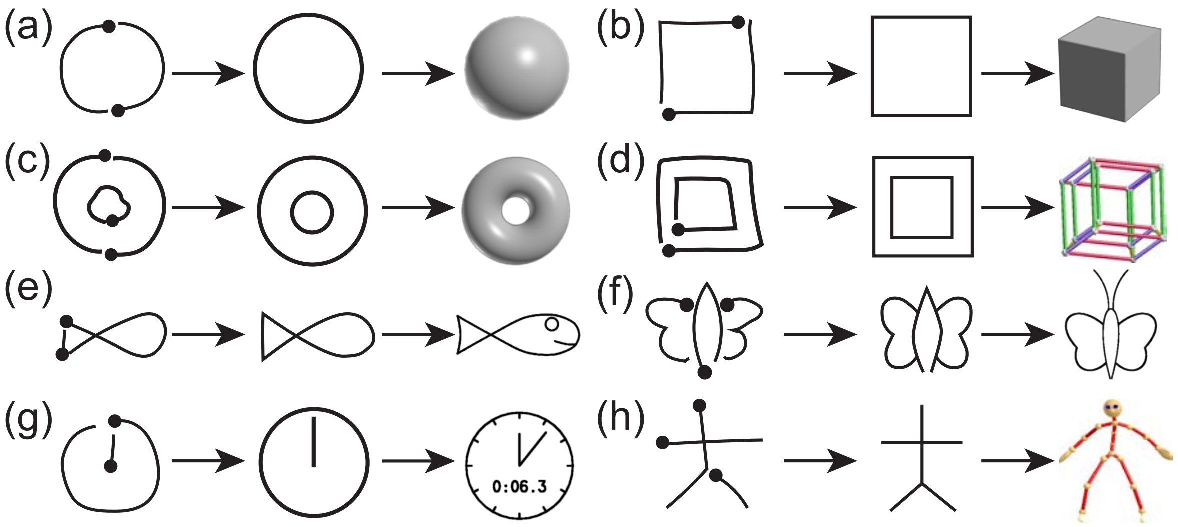


Figure 5.1: Examples of sketch recognition and object/animation generation in Chalktalk. Each subfigure shows three parts: 1) the raw sketch. Black dots indicate the starting positions of the raw strokes; 2) the intermediate conversion from the raw sketch to one of the 150 vectorized glyph; 3) the resulting instantiated object or animation. The user may translate, rotate, or scale the object as well as interact with it. These examples feature the generation of a) a sphere, b) a cube, c) torus, d) a hypercube, e) an animated fish, f) a butterfly, g) a running timer, and (h) a rigged avatar skeleton.

Though Chalktalk is designed for browser platforms, it is powerful and inspired me a lot for thinking about how to enhance communication and even collaboration through immersive environments. I then connected the idea of live sketches with different platforms such as using phones as input, drawing in AR, and sketching with others in Virtual Reality (Figure 5.2). I noticed that Chalktalk is a flexible tool for presentation and I started considering how multiple people can interact with each other in immersive environments. I explored two aspects: face to face teaching and creative collaboration.

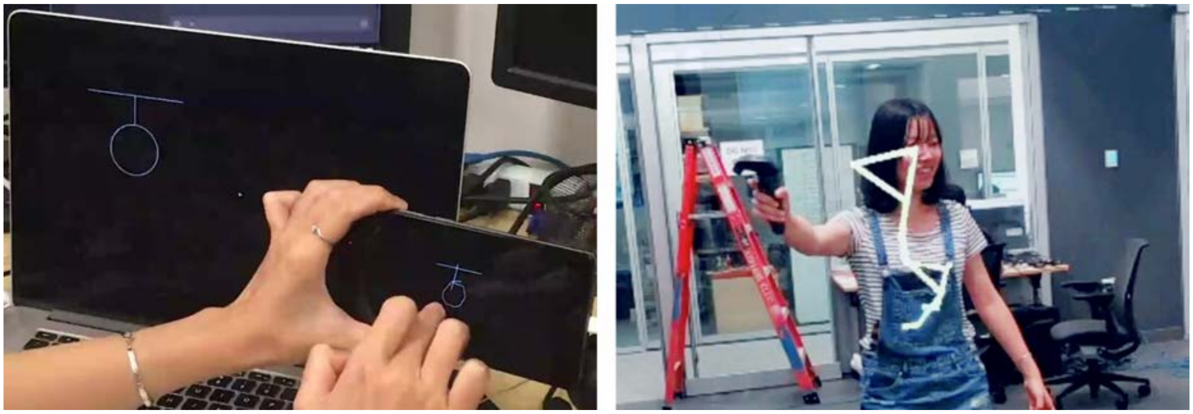


Figure 5.2: Extending Chalktalk to Handhelds and AR Environments

5.2 Introduction

Projected presentation media continue to be standard tools used for teaching. The most common and familiar software solutions, including Powerpoint and Sharepoint, are modeled after the slide-projector show, and provide a storyboard-like format, allowing for sequential text, images, and pre-defined animation sequences to be viewed one step at a time. Other options such as Prezi [Perron and Stearns 2010] additionally allow for branching sequences. These media usually supplement an oral presentation, which the speaker can perform with some flexibility for improvisation. Since traditional slideshows are sequential, they might discourage the speaker from conducting a non-sequential, more flexible presentation. If members of the audience ask questions whose answers don't lie in the slides, then the lecturer might need to abandon the slide presentation and rely entirely on speech, hand gestures, and sometimes reference objects. This becomes particularly problematic when the concepts under discussion possess dynamic behaviors or require live modifications and interactions that cannot be anticipated before the presentation is prepared.

Designed in-part to overcome these issues, Chalktalk supports not only arbitrary sequencing of content but also the creation of interactive simulations that can be combined arbitrarily as well. However, the result is still lacking in terms of presenter-audience engagement due to the way the presenter, audience, and projected presentation are located and oriented. The typical triangular format–where the presenter stands in front or to the side of the projected content towards an onlooking audience–is suboptimal. It may lead the presenter to block the view of the content in many cases, and it requires the presenter and audience to divide their attention between each other and the presented content. An alternative format would facilitate face-to-face interactions and help participants to better focus on the presenter's movements and gaze, as well as the content, with less need for attention switching.

In previous research, systems have been implemented using face-to-face interaction to improve collaboration [Ishii and Kobayashi 1992; Harrison et al. 1995; Ishii et al. 1993], communication [Otsuka 2016] and other interactions [Heo et al. 2014], and recorded lectures. Simulation of face-to-face interactions is also used in teleprompting systems, which allow a speaker to gaze at the camera naturally while reading from a projected script. Our particular focus has been to investigate the effectiveness and configuration of face-to-face interactions with respect to learning. To explore the extent to which these interactions may improve learning and engagement in the context of a lecture, we have developed a Mixed Reality (MR) platform designed for learning in a face-to-face environment. We conducted a user study in which we used projected Chalktalk and our MR system to present equivalent lessons on matrix transformations for computer graphics.

5.3 Related Work

CollaboVR is a framework to assist communication in collaboration. By definition, communication is the act of expressing and understanding among a group. Similarly, *sensemaking* is the understanding of the meaning of a communicative action [Paul 2009]. Sensemaking is a widely researched concept in information visualization. Dervin [Dervin 1992] describes sensemaking as using ideas, emotions, and memories to bridge a gap in understanding in a group. Learning how collaborative sensemaking is supported through different design considerations is very useful for multi-user communication. In this section, we first introduce collaborative sensemaking approaches. We then summarize how workspace awareness has positive effects on collaboration and how previous studies enhance workspace awareness. Last, we introduce immersive collaboration and communication and assess their advantages and limitations.

5.3.1 Collaborative Sensemaking

Prior arts have researched sensemaking [Lu et al. 2018] in HCI and computer-supported collaborative work (CSCW) area [Albolino et al. 2007; Billman and Bier 2007; Landgren and Nulden 2007; Paul 2009]. Given that sensemaking involves data analysis [Yi et al. 2008], different designs of 2D displays and digital tabletop are frequently discussed. Prior work has shared two observations. First, large and shared displays have been shown to benefit sensemaking groups in several contexts. Paul and Morris [Paul and Morris 2009] designed CoSense with a shared display, conducted an ethnographic study, and examined to support collaborative sensemaking. Vogt *et al.* [Vogt et al. 2011] found that the large display facilitated the paired sensemaking process, allowing teams to spatially arrange information and conduct individual work as needed. Moreover, multiple digital tabletops were used for sensemaking tasks [Isenberg et al. 2012; Morris et al. 2010]. Second, personal displays may lead to decreased collaboration in co-located settings [Chung et al. 2013; Wallace et al. 2009]. When designing CollaboVR, we considered the idea of "multiple" displays, displays with "different" angles, as well as adding "personal" displays into the mix, which leads to the design of different input modes and the placement of visual aids.

5.3.2 Workspace Awareness

Workspace awareness is the collection of up-to-the-minute knowledge a participant has of other participants' interaction with the workspace [Gutwin and Greenberg 1996]. It includes the awareness of others' locations, activities, and intentions to the task and to space. Maintaining workspace awareness enables participants to co-work more effectively [Gutwin and Greenberg 1998, 2002]. Workspace awareness plays a crucial role in simplifying communication, taking turns, and action prediction [Gutwin and Greenberg 2002]. Thus, maintaining and enhancing workspace awareness is beneficial to collaboration [Piumsomboon et al. 2019].

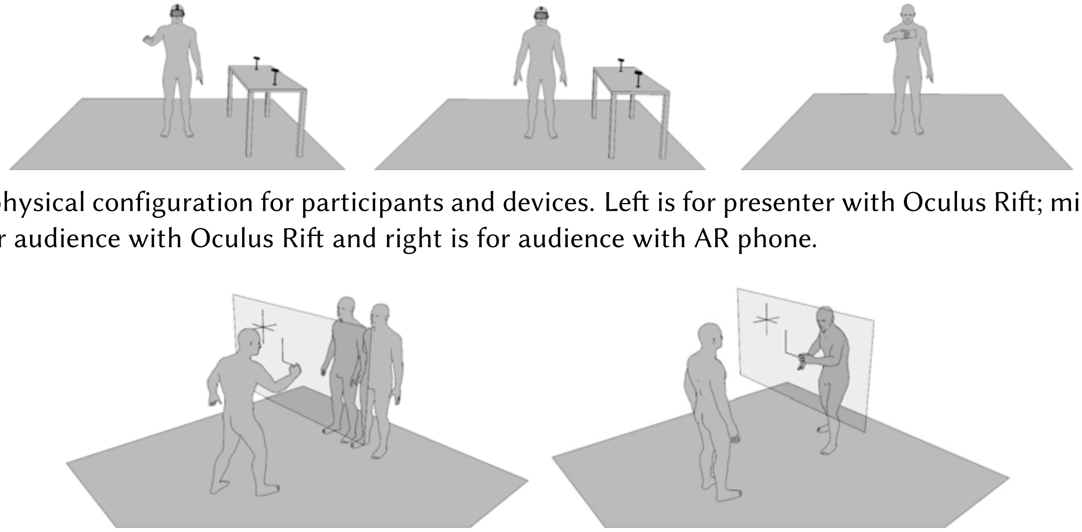
One trend is the use of see-through displays for distributed collaboration. The idea started with Tang and Minneman, who designed VideoDraw [Tang and Minneman 1990] and VideoWhiteBoard [Tang and Minneman 1991]. Both were two-user experiences. On each side, a camera was placed to capture the local user and the drawing. A projector was attached to present the remote user and the drawing. ClearBoard [Ishii and Kobayashi 1992] extended the idea and used digital media and monitor. Similarly, KinectArms [Genest et al. 2013] used a tangible tabletop as the media and rendered the arm of the remote user for mixed presence. Furthermore, Li *et al.* [Li et al. 2014] developed FacingBoard with two-sided transparent displays. Analogous to ClearBoard, the entire upper-body was displayed to other participants so gaze awareness was supported. To maintain gaze interaction, FacingBoard reversed the graphics on the display. Consequently, column-sensitive content, such as text and maps then became incorrect. To solve this problem, FacingBoard selectively flipped the column-sensitive content individually and adjusted the content position. However, when people pinpointed a specific sub-area within the content, the gaze and the place being pinpointed were inconsistent for both users. Considering flipping the content, Bork *et al.* [Bork et al. 2017] showed that the flipped version of Magic Mirror has better usability than the non-flipped version. In our system, we proposed different user arrangements to enhance workspace awareness, from which, there is a similar face-to-face experience. Differently, we manipulate the users' locations to maintain gaze awareness rather than flipping the content. That keeps the content in the original and correct format. Meanwhile, we support collaboration with more than two people.

5.4 Scenario I: Immersive Teaching

Our system supports not only VR headsets but also AR devices since our system does not require much physical space. The experience can be run anywhere – at home, at the office, or in a park. We chose the Oculus Rift as the VR headset for easy setup, and the Google Pixel phone as the AR device for an outdoor scenario. We use Unity as our development platform because it can readily accommodate executables for different hardware, with no changes required to scene design. Figure 5.3(a) shows the physical location of participants and the presenter for different devices. Here we assume that the Oculus sensors are placed on a table in the front. We have users stand aside the table facing parallel to the side of the table. The users can walk around the content, which is displayed on a plane perpendicular to the table. Figure 5.3(b) shows the immersive environment for both the presenter and the audience. The presenter and the audience are located on opposite sides of the presentation content, looking at each other face-to-face. We achieve this with a virtual mirror effect, which we describe in more detail in the following section. Here we define both presenter and audience as one category of the role. We render remote users with different roles in a mirrored way according to the content. In the following section, we discuss our rationale and implementation for this face-to-face configuration.

5.4.1 Face-to-Face Presentation

Two main approaches to the presentation are used widely: (1) the blackboard/whiteboard and (2) the projected slideshow. In the case of blackboard/whiteboard presentations, the audience usually faces the board directly, and the presenter stands at or near the board at all times. Standing at an angle to the board and audience, the presenter cannot simultaneously focus on drawing the presentation content and maintaining eye-contact with the audience [Tan et al. 2010]. As a result, the presenter must change focus repeatedly, and the audience, too, cannot maintain eye contact with the presenter when looking at content on the board. There always exists a context switch



(a) physical configuration for participants and devices. Left is for presenter with Oculus Rift; middle is for audience with Oculus Rift and right is for audience with AR phone.

(b) MR view for presenter and audience. Left is the presenter view. Presenter can see all audience; Right is the audience view. And only presenter is visible to him.

Figure 5.3: Overview for immersive teaching

to and from the board that interrupts interaction between presenter and audience. Sometimes the presenter might also obstruct the board [Fuhrmann et al. 2001] on at least one side while drawing. The multiple cameras or displays that conferences such as TED [Tan et al. 2010] afford are unlikely to be available for the typical classroom setting, and they must be configured ahead of time. For the classroom setting, it is possible for the audience to become lost when the presenter describes something being drawn while blocking that drawing from view. By the time the view of this new drawing becomes clear, some in the audience might have lost track of the argument.

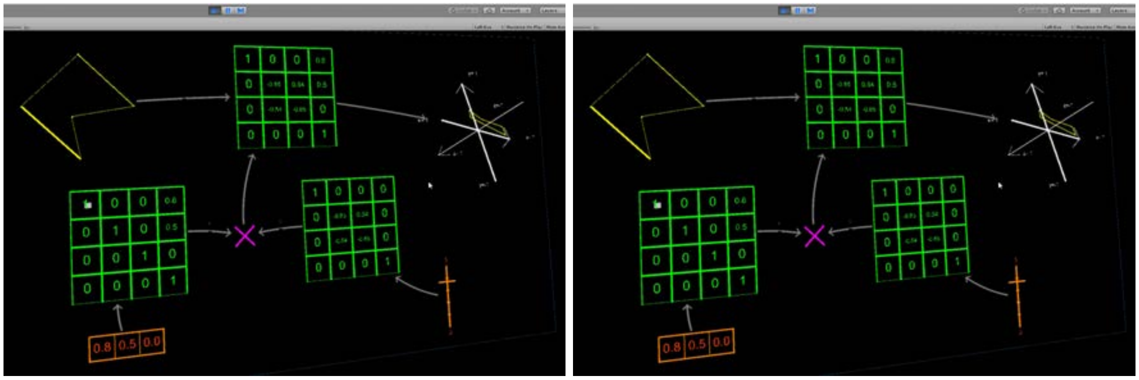
For the projected slideshow case, the presenter, audience, and board typically have the same spatial configuration as in the blackboard/whiteboard case, with the difference that slideshow content is created and sequenced prior to the presentation, and the presenter is less likely to physically block the board now that the content need not be drawn. Predefined content might save time and reduce the number of context switches for the presenter, but the presentation is fixed to that predefined content. If clarification becomes necessary during a lecture, the presenter must fall-back to drawing on a whiteboard or using spoken word descriptions. To follow the lecture, the audience must still alternate its focus between presenter and content or choose not to pay attention to the presenter, thereby reducing emotional engagement.

We render Chalktalk's sketches to a shared infinite transparent board, (we will call this the "MR board"). We found that face-to-face interactions are important to the audience [Ishii and Kobayashi 1992; Tan et al. 2010; Vertegaal 1999]. Yet if two people were to stand physically across from one another with a pane of glass between them, text on that pane of glass would appear backward for one of them. Therefore, inspired by previous work on "Clearboard", we mirror the presenter's view of students, and vice versa, such that each appears to the other left-right mirror-reversed on opposite sides of the MR board. In this way, gaze direction is preserved, allowing the presenter and audience can establish eye-contact while appearing to look at the same objects with text non-reversed for both. Since the transparent MR board lies between presenter and audience, the content itself is not blocked.

5.4.2 One-to-One Experience

We also wanted to explore how immersive one-on-one interactions might make presentations more engaging and effective for learning. In this experience, the teacher should be aware of all students, but each student should be aware of only the teacher. Thus, we built our system to allow for a multi-user presentation scenario in which one presenter addresses multiple (remote or local) audience members at the same time, while audience members are shown only the teacher. To clarify the implementation, only the presenter's avatar is rendered for each audience member. The presenter, however, sees all audience members' avatars rendered, and audience members can provide visual feedback to the presenter via body language (We send each audience member's local transform information to the presenter over a network.) All participants can also hear the same audio and communicate over a group call. This means that although audience members cannot see each other, they can still interact verbally–for example, when asking the presenter questions that everyone might want to be answered.

5.5 Evaluation I: Matrix Lecture



(a) Introduction to matrices

(b) Composition of matrix transformations

Figure 5.4: Experiment for introduction to matrices and composition of matrix transformations.

We conducted a subject-specific controlled experiment by giving a presentation using both projected Chalktalk and our system with Oculus Rift. We recruited 8 participants (P1-P8). The participants are required to have taken a linear algebra class before. The participants (50% female) are between the ages of 22 and 26 ($M = 23.71$, $SD = 1.50$) and come from various countries. According to the answers to the pre-screening questionnaire, 100% have tried VR before. All participants had also seen Chalktalk previously. This helps reduce the "novelty effect" of both presentation formats. We are particularly interested in user interview feedback for our study. We refer to this feedback as part of our results.

5.5.1 Study Design and Task

To evaluate our system's effectiveness for learning, retention, and level of engagement, we chose 4x4 matrix transformations as the presentation topic. We focused on 3D visualization of how matrices apply to objects, for those already familiar with linear algebra and matrix calculations. The content covered in the topic is translation and rotation matrices, followed by a demonstration that matrix multiplication is not commutative. We present this topic To conduct a realistic presentation, we invited a professor who teaches computer graphics to present a lecture on matrix transformation and recorded the talk as a template. The reason we chose this topic is that (1) it is complicated enough so that the presenter probably could not easily describe it via words only, and (2) it requires dynamic input to show the transformation idea in an intuitive fashion, which is a key feature of Chalktalk we wanted to incorporate, and (3) the entire presentation can be done in under 10 minutes. That is sufficient time for audiences to experience the corresponding platform along with the learning experience. Figure 5.4 shows two parts of the lecture. Figure 5.4(a) shows how transformation matrix is applied to geometry and Figure 5.4(b) shows a matrix multiplication operation and how varying the order of matrix multiplication impacts the result.

To evaluate our system using projected Chalktalk, we included some specific activities during the experiments for both platforms. The presenter in both environments (1) made gestures during the presentation to draw the audience's attention, (2) moved the cursor to point at some part of the content to see to what extent the audience was able to follow, (3) used deictic words [Hickmann and Robert 2006] to see to which extent the audience could follow those, and (4) used Chalktalk's pan operation to shift the entire view to learn how shifting content off of the visible projected display area influenced the experience. In contrast, in VR Chalktalk, because it models an "infinite screen", even when content is panned, all of the content still remains in view.

5.5.2 Results

Awareness Results. First we check the awareness of the presenter in VR and in reality (M_{VR}=6.17, SD_{VR}=0.083, $F(2,8)$=0.172, p>0.5). It turns out that the awareness of the presenter does not vary greatly when comparing the experience in VR and in reality. Although each environment has a different spatial configuration (face-to-face in VR, triangular audience/presenter/board relationship in reality) the audience is always aware of the presenter's presence, never "tuning him or her out" completely. In other words, the location of the presenter has very little impact on how aware and focused the audience is on the presenter.

Focus Results. For each environment (VR and reality), we check the extent to which the audience finds it easy to (1) focus on the presenter, (2) focus on the presentation content, and (3) shift focus between presenter and presentation content. Looking at the results for (1) ($F(2,8)$=0.63, p>0.1) and (2) ($F(2,8)$<0.01, p>0.1), the difference in feeling when comparing the environments is small. P7(F) mentioned that the cursor on projected Chalktalk (displayed on a large monitor) is too small for her to follow and P1(M) emphasized that in VR the avatar's drawing hand fulfilled the role of a huge cursor for him, which helped him stay fully focused on the content. The result of frequency of shifting focus is ($M_{reality}$=5.33, $SD_{reality}$=1.21, $F(2,8)$=11.36, $p = 0.02$). From that, we can tell that face-to-face design has a strong impact on how actively the audience shifts focus. In the interview, P5(F) described switching focus between presenter and content much more in the real environment. Multiple participants mentioned that the face-to-face configuration helped them concentrate.

One-on-one Experience. For the VR environment, we asked to what degree there was a feeling of one-on-one interaction, resulting in (M=6.17, SD=0.75). We also asked to what degree the audience member is aware of the other audience member: (M_{VR}=1.5, SD_{VR}=0.84, $F(2,8)$=30.94, p<0.01). In VR, although two audience members cannot see each other, we support group audio so both of them can speak to the presenter. All participants tried to speak to the presenter during the experiments in VR. The audience felt a sense of being given a one-on-one presentation when only the presenter was speaking (P2; P8, F), and they didn't encounter distractions in VR since the other audience member provides only audio feedback–no visual representation. This suggests that a one-on-one feeling helps the audience concentrate on the presentation.

Enjoyment and Learning Results From the questionnaire, we see that there is no large difference in the level of learning (M_{VR}=6.67, $M_{reality}$=6.5) and enjoyment (M_{VR}=6.67, $M_{reality}$=5.67) between VR and reality formats. P5(F) suggested that if the presentation platform itself is interesting, then it will help encourage the audience to be more focused and have a positive impact on the learning experience. That means that (1) she thinks that the VR setup is more intuitive and interesting to her and (2) that she can focus on the content more easily in VR than in other setups.

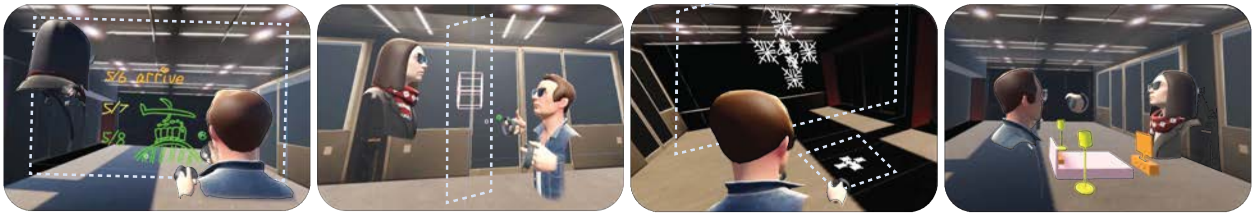
5.6 Summary

It turns out that our system significantly decreases focus shift during the presentation, which may help participants concentrate on the content. Also, we learned that participants barely notice the existence of the other audience members, and received the experience as a one-on-one presentation with the presenter, which helped them to better concentrate on the presented material. We also confirmed some properties in MR are good for presentations, such as its lack of display space restrictions. The content and appearance of the participants, as well as customized support for different students, were all found to be important design factors in this scenario.

Next, we enlarge the scope that could bring benefits to collaboration for general use. As we know communication is happening every day and everywhere. However, it is not always very effective. For example, sometimes, we need to sketch on whiteboards for better clarity, but someone's line of sight may be blocked. We realized that sketching, one of the most natural and fun ways to express ourselves, has rarely been explored in collaborative VR. Additionally, it is an open question what is the best layout and interaction mode for creative collaboration: An in-air shared canvas between users? A whiteboard in front of users? A notebook or a tabletop to be shared by users? Motivated by these alternate metaphors, we investigate the following research questions: What if we could bring *sketching* to *real-time* collaboration in virtual reality? If we can convert raw sketches into *interactive animations*, will it improve the performance of remote collaboration? Are there best *user arrangements* and *input modes* for different use cases, or is it more a question of personal preferences?

To answer these questions, I have developed an end-to-end system for both distributed and co-located multi-user communication in virtual reality. The system employs a cloud architecture in which applications such as Chalktalk [Perlin et al. 2018a] (a software system to convert raw sketches to digital animations) are hosted on the server. This architecture allows geographically dispersed clients to talk with each other, sketch on virtual sketching boards, and express ideas with interactive 3D animations with low-end VR headsets. Furthermore, CollaboVR allows for real-time switching between different user arrangements and input modes. Whether the user intends to draw on a notebook, sketch in the air, or have a discussion in front of a whiteboard, CollaboVR can instantly and seamlessly switch context to support the user arrangement and input mode of choice.

5.7 Scenario II: Creative Collaboration



(a) Discussing travel schedules in *integrated layout* with remote participants. (b) Presentation on the topic of hyper dimension in *Mirrored layout*. (c) Sketching a baroque-style pattern in *projective layout* to remote users. (d) Collaborative design session of furnitures and apartment arrangements.

Figure 5.5: CollaboVR is a reconfigurable VR framework that combines the abilities of animated sketching, collaborative scene editing, and multi-user communication in real-time. We showcase four use cases in custom *layouts*: (a) shows an *integrated* layout of a business meeting, (b) shows a *mirrored* layout of a math class presentation, (c) shows a third-person perspective of the *projective* layout where the user draws at hands and projects his sketches to remote participants on the shared interactive board, and (d) shows two roommates discussing the apartment design. Please refer to the supplementary video for live demos.

Our overarching goal is to propose a reconfigurable framework for creative collaboration in VR, which can adapt to different use cases and optimize virtual spaces depending on the selected task. We restrict our scope to teamwork with whiteboards and visual information. We next describe our use cases and system architecture.

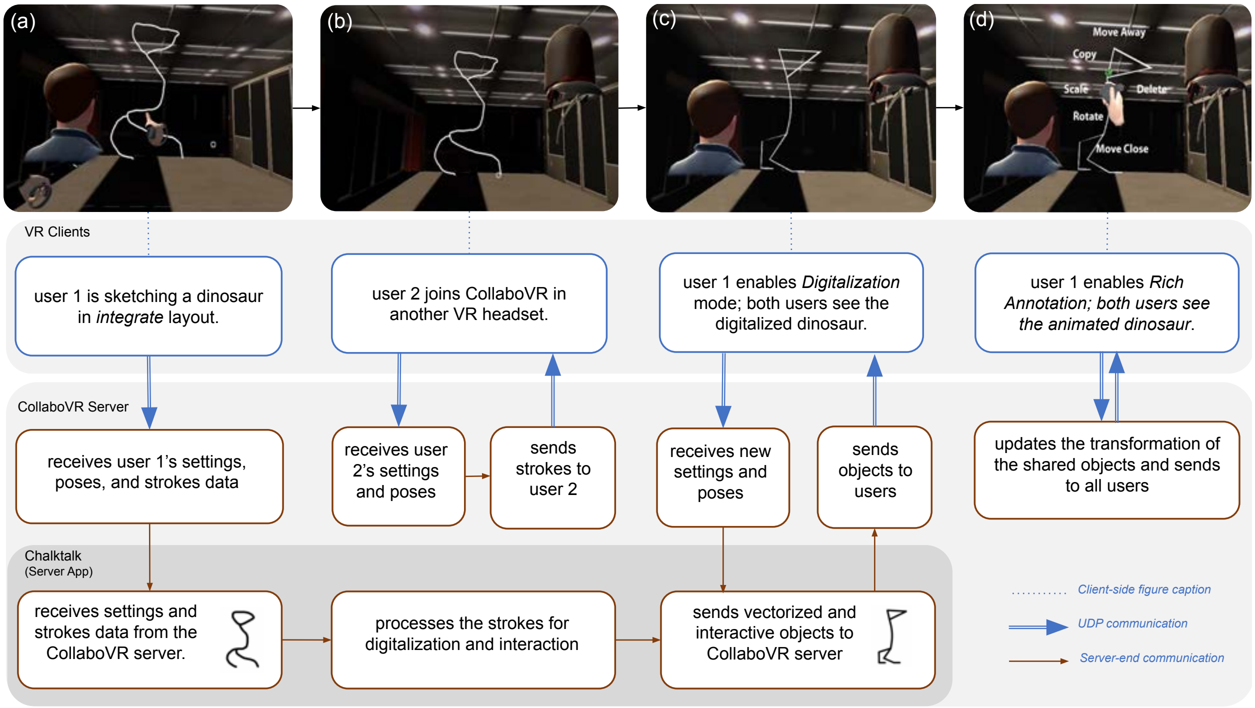

(a)
(b)
(c)
(d)
Move Away
Copy
Scale
Delete
Rotate
Move Close
VR Clients
user 1 is sketching a dinosaur in *integrate* layout.
user 2 joins CollaboVR in another VR headset.
user 1 enables *Digitalization* mode; both users see the digitalized dinosaur.
user 1 enables *Rich Annotation; both users see the animated dinosaur.*
CollaboVR Server
receives user 1's settings, poses, and strokes data
receives user 2's settings and poses
sends strokes to user 2
receives new settings and poses
sends objects to users
updates the transformation of the shared objects and sends to all users
Chalktalk (Server App)
receives settings and strokes data from the CollaboVR server.
processes the strokes for digitalization and interaction
sends vectorized and interactive objects to CollaboVR server
Client-side figure caption
UDP communication
Server-end communication

Figure 5.6: The workflow: (a) As user 1 sketches in CollaboVR, the server receives the aggregated data of settings, poses, and strokes and sends the strokes data to Chalktalk for further processing; (b) When user 2 joins CollaboVR, the server broadcasts the poses from user 1 as well as the latest stroke data so that both users see the sketches and each other; (c) After user 1 triggers the *Digitalization* mode and notifies the CollaboVR server, the server queries Chalktalk. In less than 16 milliseconds, Chalktalk converts the strokes into interactive objects. Then both users see digital objects (in this case, a triangle and several spline curves) from the CollaboVR server; (d) When user 1 performs *Rich Annotation* on the sketch, the CollaboVR server alone handles commands for scene editing tasks.

We envision the following potential use cases for CollaboVR.

Travel planning and brainstorming. CollaboVR can be used for trip schedule as presented in Figure 5.5(a). Multiple remote users are rendered as virtual avatars in front of a large virtual interactive board. With freehand drawing, users can write and draw their desired travel plans and coordinate with friends via both audio communication and sketches. When they have different ideas, users can easily duplicate the current interactive board and iterate on the prior one to express new alternatives.

Interactive lectures. CollaboVR can also be used for interactive classes as shown in Figure 6.3(b). In this case, CollaboVR places the presenter and the audience on opposite sides of the interactive board. Sketches are shown identical to both the audience and the presenter so that the presenter and the audience observe the same scene. With face-to-face remote communication, the presenter may pay more attention to the audience's focus, while the audience can simultaneously follow the presenter's gestures and content.

Presenting live designs on a sketchpad. Writing directly on a whiteboard is not always preferred in creative collaboration sessions. Many users feel more comfortable writing on a notepad or tabletop while sitting in a chair. Hence, we enable CollaboVR to support a projection input mode as shown in Figure 5.5(c). In this example, the lead designer can focus on sketching a baroque pattern on a small, flat, private interactive board. The experience is similar to drawing on a digital tablet with a pen while the contents will be projected to the large, shared interactive board to other audiences. Upon finishing, the lead designer may look at the audience and ask for questions and suggestions. Other participants can contribute by pointing or sketching onto the projected design draft.

Designing spatial layout. CollaboVR can also help with designing spatial layouts, especially in 3D. Imagine that a user has just moved into a new apartment and needs to remotely discuss the placement of the furniture with other roommates. As Figure 5.5(d) demonstrates, the user can draw furniture with a combination of primitive 3D objects and place them directly at preferred locations. Spatial layout is difficult to describe clearly through words and gestures, and it often requires freehand drawings, multiple iterations, and multiple perspectives in 3D. CollaboVR satisfies users' needs by offering them a rich set of interactive tools and real-time sketch-to-object techniques via cloud-apps.

5.7.1 System Architecture and Workflow

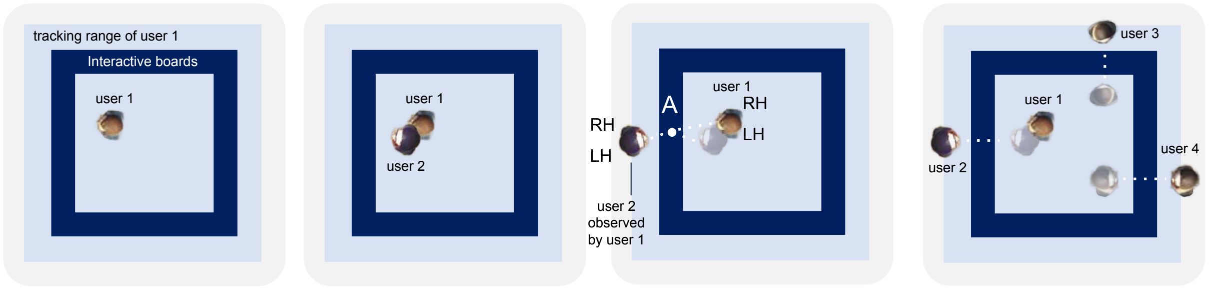
CollaboVR aims to offer a reconfigurable architecture for creative collaboration in VR with lightweight software on the client-side and low-latency services on the server-side. Hence, we leverage a cloud-based architecture where the computational expensive applications are hosted on the servers and the rendering results are streamed to all clients.

As a proof-of-concept, we employ Chalktalk [Perlin et al. 2018a] as the server application to enable creative collaboration in VR. While there are many smart sketch-based online software programs – such as Autodraw [Google 2019], sketch2code [Lab 2019], and Miro [Software 2019] – that can assist creative collaboration, Chalktalk is open-source software with a rich set of sketch-based communication language and digital animations. It allows a presenter to create and interact with animated digital sketches on a blackboard-like interface. We chose to use Chalktalk because it is an open-source platform, so we can easily define the data-flow between the application and CollaboVR.

We designed an extendable protocol in the CollaboVR framework so it can work with other applications as long as the input and output are accessible. The protocol serializes input and display data from each user, routes that data through a network, and then de-serializes and interprets the data to correctly render the results into graphics.

The CollaboVR server is written in Node.js and C#. It synchronizes data across devices and supports custom data formats. For CollaboVR, we have two kinds of information: rendering data and user data. Figure 5.6 demonstrates the workflow of ColalboVR. For rendering data, we first pass the user input from each client to the server. Then, the server transmits the user input together with its user identifier to the application. Next, the server receives the serialized display data from the application (Chalktalk). Finally, the server broadcasts the display data to each client for rendering. For user data, we broadcast the user's avatar, poses, and audio stream to each client after it has been received.

To unleash the users' creativity, we design "Rich Annotation" mode to empower CollaboVR clients to manipulate sketches and objects. After the clients receive and render the display data from the application, the display data are considered as interactive objects in a 3D world. This manipulation includes duplication, linear transformation (rotating, scaling, and translation), deletion, and colorization.



(a) User 1 joined CollaboVR; walking range is limited within the interactive boards.

(b) After user 2 joined CollaboVR, side-by-side arrangement may cause occlusion or collision.

(c) Face-to-face arrangement solves this issue by mirroring user 2 by the shared interactive board.

(d) Face-to-face arrangement enables one-to-many communication and reduces visual clutter.

Figure 5.7: Comparison between *side-by-side* and *face-to-face* arrangements. (a) shows one user in CollaboVR. Interactive boards are depicted as dark blue rectangles. (b) shows two users in the side-by-side arrangement. This arrangement is intuitive to users and supports side-by-side whiteboarding tasks, but may cause occlusion or collision of virtual avatars. (c) shows how a face-to-face arrangement can solve this problem. For each user, the face-to-face arrangement mirrors the other user, so direct eye contact is preserved and both users can see each other while sketching on the same interactive board. The spatial direction remains the same, see 'LH' and 'RH' indicators of user 1 and user 2 observed by user 1. (d) shows an extended version with four users from user 1's perspective. Each user sees the other mirror-reversed through their respective boards. Compared with the side-by-side arrangement, our face-to-face arrangement reduces visual clutter while maintaining eye contact.

5.7.2 Custom Configurations

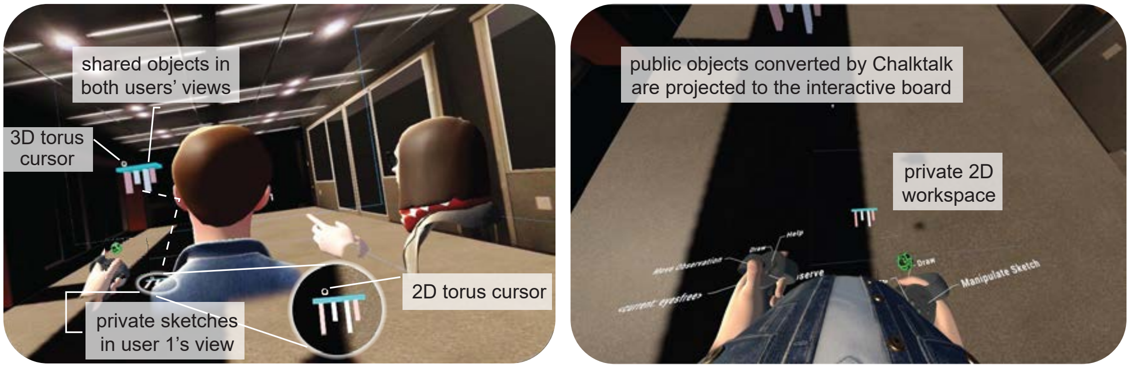
As motivated in the Introduction, we designed CollaboVR as a reconfigurable framework to investigate the best configuration for creative collaboration tasks. Previous work has great insights on one specific user arrangement or input mode. We investigate three user arrangements (integrated, mirrored, and hybrid) and also offer two input modes (direct and projection). "Projection mode" is designed to see whether it is more effective for expressing ideas in remote presentations. Inspired by prior art in workspace awareness [Gutwin and Greenberg 1998, 2002], we focus on maintaining and enhancing workspace awareness, to empower participants to work together more effectively. CollaboVR allows users to alter their views of other participants. In other words, they can manipulate the spatial layout by which they see other users.

Concluded by previous work on collaborative sensemaking [Isenberg et al. 2012; Morris et al. 2010; Paul 2009], we notice that multiple and shared large displays are useful for collaborative work in terms of 2D information. CollaboVR is an immersive 3D graphics world. Instead of "display", we set up multiple "interactive boards" in the virtual environment.

User Arrangements. We offer three user arrangements for CollaboVR: (1) side-by-side, (2) face-to-face, and (3) hybrid arrangement. The side-by-side arrangement places each remote user into a shared virtual space, which is defined within the tracking range of the VR headset as shown in Figure 5.7(a). The side-by-side arrangement enables multiple users to collaborate side-by-side in front of the same interactive board. All users may focus on the contents during the creative collaboration. However, two user avatars may be occluded with each other as illustrated in Figure 5.7(b).

The face-to-face arrangement solves the occlusion issue by mirroring all the other avatars' locations to the other side of their currently activated interactive board. In Figure 5.7(c), user 1 enables the face-to-face arrangement so user 2 in user 1's view is mirrored to the other side of the left interactive board which user 2 is looking at. Now let's take a look at the gaze interaction. Spot A is the same content that both users are looking at. After the mirroring operation, the gaze direction of users is maintained. Moreover, users are aware of each other's focus while gazing at spot A at the same time. In contrast to FacingBoard [Li et al. 2014], we did not mirror-reversed the content so the content is still correct to each viewer. We then consider how spatial instruction looks like. User 2 with transparent shading indicates the original position of user 2. From Figure 5.7, we know that user 2 is on the left side to user 1 originally. Equivalently user 1 is on the right side of user 2. After enabling face-to-face arrangement in user 1's perspective, the spatial relationship remains the same for all the users. The face-to-face arrangement is like a mirror. Users only need to consider the spatial relationship from their own perspective, see the left hand (LH) and right hand (RH) indicators in Figure 5.7(c) for user 1 and user 2 observed by user 1. In this user arrangement, the users can see each other for better workspace awareness. Figure 5.7(d) illustrates the multi-user scenario when users are looking at different interactive boards. Compared with the integrated layout where every user is restricted within the shared virtual space, our mirrored layout greatly reduces the visual clutter and maintains users' eye contact.

The hybrid arrangement inherits the "teaching in a classroom" metaphor, where the teacher uses the face-to-face arrangement to observe students, and the students use the side-by-side arrangement for classmates and a face-to-face arrangement for the teacher. We envision that this arrangement may be useful for online education with a large audience.


shared objects in both users' views
3D torus cursor
2D torus cursor
private sketches in user 1's view
public objects converted by Chalktalk are projected to the interactive board
private 2D workspace
Help
Draw
Move Observation
<current: eyesfree>
Draw
Manipulate Sketch

(a) third-person perspective of the projection mode with two users.

(b) first-person perspective of the presenter modifying private sketches.

Figure 5.8: Projection mode. (a) demonstrates user in blue drawing a table in projection mode from a third-person perspective. There is a private sketch that only the person who is drawing can see. It is laid out in 2D at the user's waist height, meanwhile a 3D object is displayed on the interactive board for all the users to see. (b) shows the first-person perspective when the user looks down and creates his 2D sketch.

Input Modes. Motivated by the two metaphors of writing on a whiteboard and sketching in a notebook, we offer two input modes in CollaboVR to support different use cases: direct mode and projection mode. Direct mode adapts the metaphor of writing on a whiteboard (Figure 5.5(a)). This may be best used where the user experience is similar to a brainstorming or interview session in the meeting room.

In addition to the direct mode where the user sketches on the interactive board, CollaboVR enables projection mode, where the user may sketch on a private workspace at the hands and project the contents onto the shared interactive board. We present both a third-person and a first-person perspective of the projection mode in Figure 5.8. The private workspace is placed at an approximately 1-meter height, lower than users' hands, so the drawing won't be displayed above users' arms whether sitting or standing. For the other users who are not sketching, the content is duplicated and rendered on the shared interactive board, see Figure 5.8(a). By doing this, we avoid a situation whereby the content is not readable for all the users around a table. When the user is writing in the private workspace, he/she is free to look at the personal workspace or the shared interactive board (see Figure 5.8(b).) Moreover, content on the private workspace is different from the content on the shared interactive board in two ways: scale and dimension. Given that the reach distance when writing on a private workspace is smaller than on the shared one, we adjust the scale of the private workspace. Regarding the dimension of the content on the private workspace, we squeeze the content and render the content in 2D. (See how the table looks in Figure 5.8.) The reason we implement squeezing is that we prefer to simulate a tablet-style input and keep the designing space clean as well. To enhance the awareness of where the user is writing, we render the projection point of the user's controller as a 3D/2D cursor (Figure 5.8).

5.7.3 Technical Implementation

We implement CollaboVR with an extendable networking protocol, a calibration approach for co-located users, client software for freehand sketching and object manipulation, and a server-end software, Chalktalk to digitalize the sketches and generate animations.

Networking Protocol. For each creative collaboration session (like client session or server-end application session), CollaboVR establishes a UDP network for low-latency and real-time performance. The user data and rendering data need to be transmitted every frame. The server is written in Node.js and the client is written in Unity C# and Node.js.

We defined a *synchronizable object* as an object that needs to be synchronized each frame for the client who registered it. Each synchronizable object has a label and data stream. The label is a unique id for the client to register. The data stream includes sending frequency and real-time data.

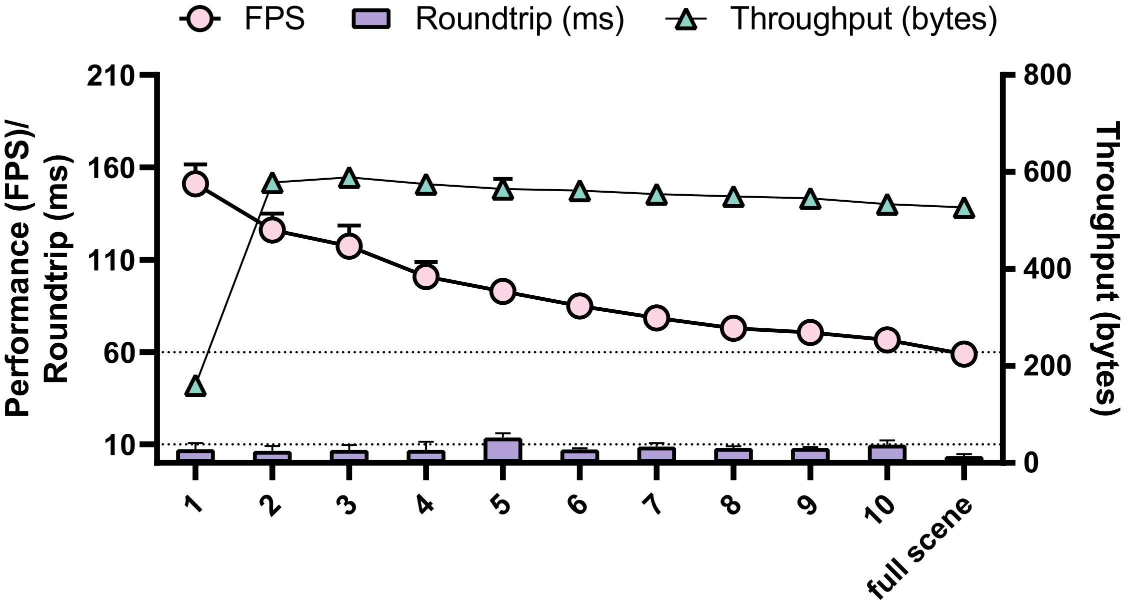
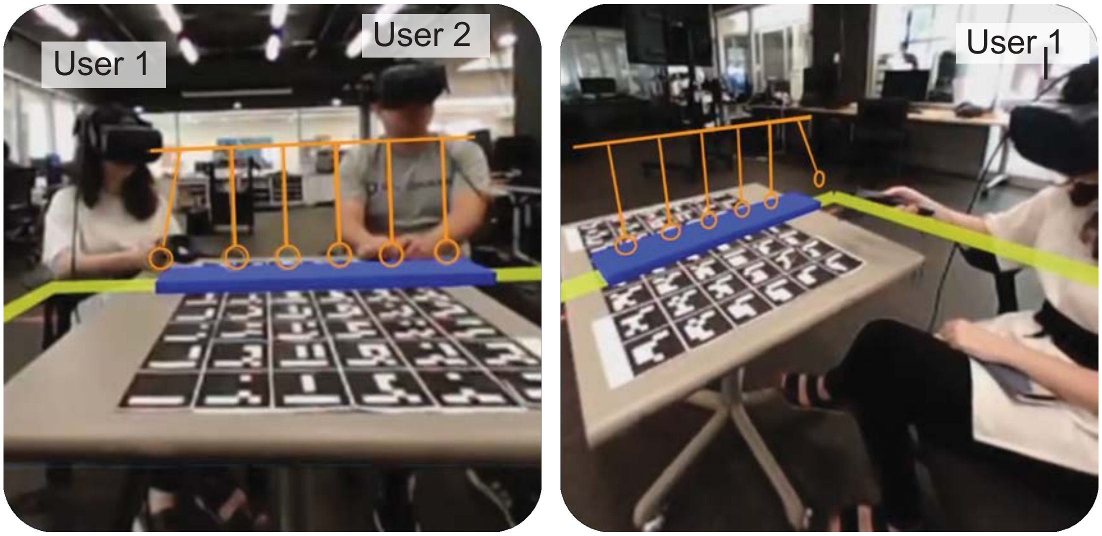


Figure 5.9: Chart of network latency and rendering performance as the number of clients ranges from 1 to 10. The last column shows the results with 10 clients as well as a fully designed living room. Networking latency remains around 10ms consistently; rendering performance drops from 160fps to 60fps with 10 clients; throughput does not change appreciably with an increased number of clients, but depends rather on the complexity of the displayed scene.

We provide two frequency values in the system: *one-time* and *per-frame*. A *one-time* synchronizable object is designed for sending commands including `join CollaboVR`, `switch to certain board`, `select objects`, etc. It does not happen for each frame. For a *one-time* object, we use a two-way handshaking metaphor. The client sends the object to the server, the server returns an object including acknowledgment back to the client, then the client deregisters the object with this local label. The *per-frame* synchronizable object includes avatar representation, the audio data, and the display data from the Chalktalk application. We design a protocol to wrap all the display data. The data protocol includes information of all the rendered lines and meshes by encoding their attributes. Each client deserializes the data from the server and renders the deserialized data as strokes or meshes. Figure 5.9 shows how CollaboVR performs with an increased number of clients. We evaluate a four-client case in user study while the system can afford at least 10 clients simultaneously. Overall, the networking latency is under 10 ms, rendering performance stays above 60 frames per second even when there are 10 clients discussing a full living room scene, and the throughput per frame is quite stable when the number of clients increases.



(a) Two users discussing an interactive Newton's cradle.

(b) First-person view of user 2 when user 1 manipulates the cradle.

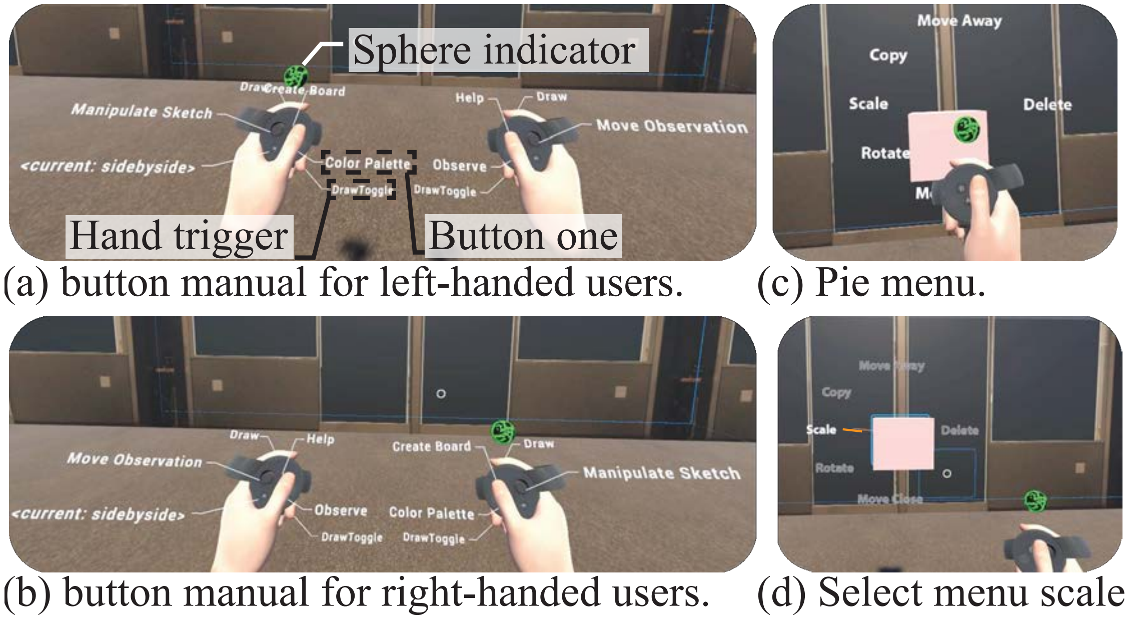
Figure 5.10: An example of our co-located user setup using HTC Vive Pro with accurate calibration. (a) shows two users discussing Newton's cradle in CollaboVR. (b) shows user 1 dragging a virtual ball to interact with objects in CollaboVR.

Calibration for Co-located Scenarios. CollaboVR works for both co-located and physically distributed scenarios. For distributed users, we simply overlap their virtual environments because they do not have any spatial relationship in reality. For co-located users, we need to carefully calibrate their relative locations, so their avatars are rendered in the same coordinate system. The key idea for calibration is that different clients should have a shared trackable proxy by their camera systems.

In Figure 5.10, we present an example with HTC Vive Pro headsets in the co-located modes of CollaboVR. We enabled the mixed-reality mode to capture the co-located user setup. The shared proxy in the Vive system is the base station. Each machine running Vive can retrieve the transformation of the base station. Because all machines (assuming N machines) have their own coordinate systems, we have N pairs of the transformation of the base station. We choose one base station as the proxy based on the unique serial number. Then, we treat the first connected client as the reference node. Later, all the following $N - 1$ clients apply the inverse matrix between the base station of the reference node and their own base station. Figure 5.10 shows user 1 drawing a physics model. Figure 5.10(a) presents the front view and Figure 5.10(b) presents user 2's view. With this co-located setup, users are unlikely to collide with each other and have occlusion.

Client Software. CollaboVR includes UI for users to convert raw sketches into digital objects and manipulate them after freehand sketching. We provide the functionality of duplication, transforming, deletion, and colorization. To achieve this, we designed a pie menu triggered by the controller. The following is the workflow for a user's manipulation: first, place the controller so it hovers over the drawing of interest; second, press the thumbstick of the dominant controller; and then, the pie menu appears as Figure 5.11(c); later, move the thumbstick to select the specific menu (see Figure 5.11(d)); afterward, apply the corresponding movement in terms of the command and release the thumbstick. The color palette is toggled by button one, illustrated in Figure 5.11(a). The user can drag the color from the palette to any drawing like world builder [reklamistcomua 2019].

The controller's trigger switches the commands of the two controllers for left-handed and right-handed users (see Figure 5.11). As the user's view might be blocked by other users' avatars, we implement a spectator mode. Users can see the view from different users in the lower right corner. To encourage all users to work on the task together, we implement a permission strategy. Only one user can draw at one time. Once the user who is drawing releases the permission, other users can grab permission to draw, see Figure 5.11(a). Deploying CollaboVR requires only a VR



(a) button manual for left-handed users.

(c) Pie menu.

(b) button manual for right-handed users.

(d) Select menu scale

Figure 5.11: User interfaces for sketching and scene editing in CollaboVR clients. (a) and (b) present the button manual for left-handed and right-handed users, respectively. A small green sphere indicates which hand is currently enabled for drawing. (c) shows the interface when the user selects the color palette function. (d) shows the interface for scene editing.

device running Unity for each client, a server machine running Node.js, and an optional router for ensuring low latency for data transmission.

5.8 Evaluation II: Furniture Sensemaking

We evaluate the interaction cycles, design variables, and collaborative effectiveness of CollaboVR through a within-subject study to answer the following research questions: how does sketching affect real-time VR collaboration; how does interactive animations impact an individual's behaviors, will it improve the performance of remote collaboration; are there best *user arrangements* and *input modes* for different use cases, or is it more a question of personal preferences? During the study, we collected qualitative feedback to gain insight into the potential benefits and impacts of CollaboVR, and quantitative data to research the most preferred layout with a collaborative design task.

5.8.1 Participants and Apparatus

We recruited a total of 12 participants at least 18 years old with normal or corrected-to-normal vision (5 females and 7 males, 1 left-handed and 11 right-handed; age range: 20 – 30, $M = 23.58$, $SD = 3.45$) via campus email lists and flyers. None of the participants had been involved with this system before. The participants have reported various VR experiences in a questionnaire (rating scale: 1 (less) to 7 (more experienced), $Mean = 4.08$, $SD = 1.83$).

We deployed CollaboVR using Unity on workstations running Windows 10 with Nvidia GTX 1060 GPU, Intel Core i7 2.80 GHz CPU, and 16GB of RAM. We used Oculus Rift CV1 with two Touch controllers. Computers were connected to the router through Ethernet cables. For the duration of the study, participants' behavior, including their interaction patterns, body language, and strategies for collaboration in the shared space were observed and recorded.

In the study, four participants were grouped as a team. We instructed each group with one training session and three design sessions to perform a collaborative design task. After the design session, the researcher conducted semi-structured interviews to obtain additional insights into the most salient elements of the users' experience, challenges, and potential user scenarios. Next, we detail the training stage, design sessions, and interview stage.

5.8.2 Training Stage

At the beginning of each study session, we first introduced the system to the participants and collected consent forms for screen recording and video recording. Next, we gave the group a 10-minute lecture on Chalktalk and taught the participants how to create freehand drawings and convert them to 3D objects.

In the next 10 minutes, participants were given a demo on how to use CollaboVR. As part of the demo, a researcher put on the headset mirrored the VR content with a regular monitor and described how to use each button as well as each function, including sketching on the interactive board, obtaining permission to draw, and manipulating drawings and objects. Afterward, all participants were placed in physically distant locations with an Oculus Rift running a CollaboVR client. We instructed the participants to try in-air sketching and object manipulation until all participants were familiar with the interaction paradigms. Finally, we put all participants into a shared virtual environment and started design sessions. Overall, the entire training session took approximately 30 minutes.

5.8.3 Design Sessions

Next, all the groups were asked to experience three 10-minute sessions in randomized orders. Each session featured a different condition motivated by real-world scenarios as follows:

C1: **integrated layout** which inherits the "physical side-by-side white-boarding" metaphor. This condition places all participants into a shared virtual space without any further arrangement. However, remote users have to rearrange their avatars to avoid visual clutter and occlusion.

C2: **mirrored layout** which inherits the "face-to-face communication" metaphor. This condition resolves the former clutter and occlusion issues by using the face-to-face arrangement as introduced in subsection 5.7.2.

C3: **projective layout** which inherits the "lecture with a presentation" metaphor. In this condition, users can draw their design in their private workspace (as explained in subsection 5.7.2) and then project it into the shared whiteboards to the audience at the opposite side. This may allow users to focus on individual designs without too much distraction from the shared whiteboards.

To explore the use of the CollaboVR system for creative collaboration in the shared virtual space and motivated by the "building block" task in Holoportation [Orts-Escolano et al. 2016], we further designed a "living room design" task. In each session, the participants were asked to design a living room containing only three pieces of furniture: a table, a chair, and a couch. To simulate conflicts and encourage discussion as in normal meetings, we asked each participant to pick one piece of furniture, sketch an original 3D design, and write down the layout of the three pieces of furniture before entering CollaboVR. We instruct the participants to be creative in the color, shape, and textures of the selected furniture. Since only three items are assigned to four participants, the participants would have to resolve conflicts and come to a consensus through CollaboVR. After the individual ideation phase, the researcher instructed each participant to wear the VR headsets, enter CollaboVR, express their original ideas, and attempt to reach an agreement for the living room design. After each design session, they took off the headset and wrote down their final decisions for the design in a text file. After a five-minute break, they entered the next 10-minute session.

5.8.4 Semi-structured Interview and Data Collection

Afterward, the researcher presented the participants with a set of statements (adapted from System Usability Scale [Brooke et al. 1996] on CollaboVR and each session on a 7-point Likert scale). Next, the researcher conducted a semi-structured interview asking about their experience, trying to gain insight into usability and use cases of the system.

We conducted one-way repeated measurements analysis of variance (RM-ANOVA) statistical analysis to examine the variations between different conditions on user preference, usability, and collaboration effectiveness for each participant, and the *task performance* for each group. *Task performance* is defined as the details of the living room design for each session. We analyzed what they wrote before and after each session by calculating the quantity of the details, such as color, shape, and texture. For example, "a *yellow triangle-based* table with *flower texture*" is counted as

3 points, "a chair with *wood material*" is counted as 1 point. The collaborative task is aiming at how participants discuss and come to a consensus of a topic requiring visual description, rather than how well their final design appears. Therefore, we observed the final design they completed, yet did not take its aesthetics into performance evaluation. The level of RM-ANOVA significance was set at $p < 0.05$.

5.8.5 Results

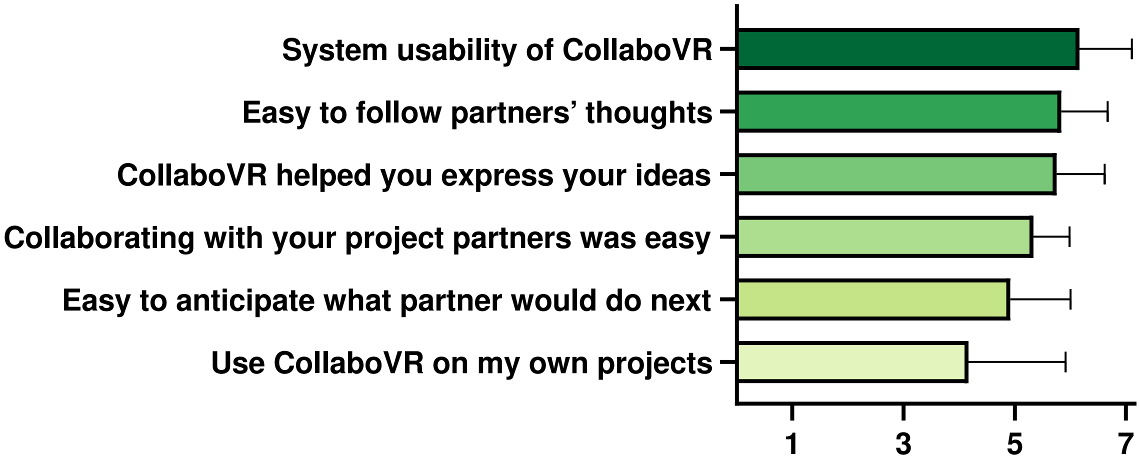
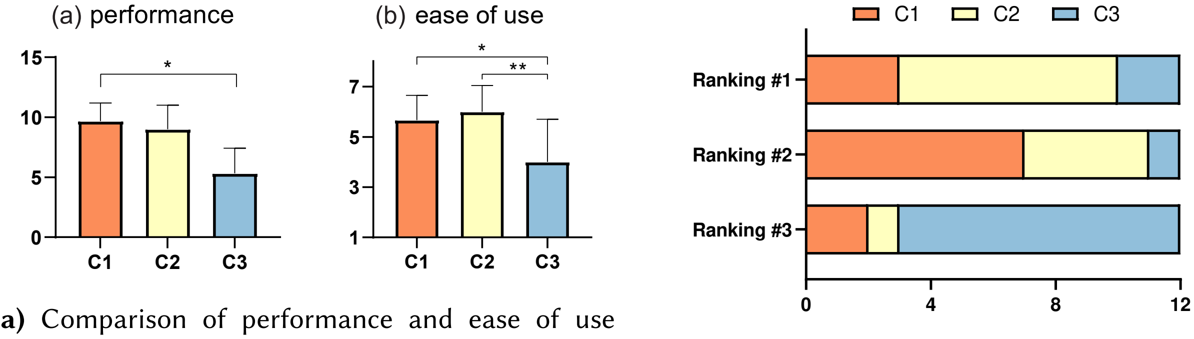


Figure 5.12: Overview of subjective feedback on CollaboVR. On the SUS, participants categorized CollaboVR as a "good and acceptable" system, $M = 6.17$. It was moderately easy to follow others' thoughts ($M = 5.83$), to express the ideas ($M = 5.75$), and to collaborate with partners ($M = 5.33$) with CollaboVR. Participants were positive about anticipating partners' next movement ($M = 4.92$) and using CollaboVR on their own projects in the future, $M = 4.17$.

In this section, we analyzed CollaboVR in general, compared each condition for individual behaviors, and evaluated the effectiveness of collaboration for three conditions. In brief, we examined that CollaboVR is helpful to express ideas with high usability. Out of three conditions, the majority of the participants preferred the mirrored layout and found it good for task completion and partner connection. Participants showed the willingness of using CollaboVR in daily life and shared the thoughts of ideal scenarios for three conditions.

CollaboVR in general. We analyzed the result of CollaboVR usability ($M = 6.17$, $SD = 0.94$), how helpful is CollaboVR to express ideas to the group ($M = 5.75$, $SD = 0.87$), and whether the participant wants to use CollaboVR in their own projects in the future ($M = 4.17$, $SD = 1.75$) (in Figure 5.12). From the observation, CollaboVR's pipeline was quickly mastered by all participants during the training session. P9(F) commented *"it is intuitive to do the drawing in 3D."*. Moreover, P11(M) responded, *"it's totally a great prototyping idea/prototyping system. Can't say it'll replace AutoCAD, but in a few years it will do that."*.



(a) Comparison of performance and ease of use among integrated layout (C1), mirrored layout (C2), and projective layout (C3). *: $p < 0.05$, **: $p < 0.01$. We found a significant difference in performance between C1 and C3; ease of use between C1 and C3, C2 and C3. $p_{\text{performance}}(C2, C3) = 0.67$.

(b) Rankings of user preferences among integrated layout (C1), mirrored layout (C2), and projective layout (C3). Mirrored layout is preferred the most for the "living room design" task.

Figure 5.13: CollaboVR Experiment Results on Performance, East of Use, and Preference.

Individual behaviors among conditions. We conducted RM-ANOVA tests to compare the effect of three conditions – integrated, mirrored, and projective layout – on ratings, how helpful for performing tasks, in-sync with other partners, connected with other partners, and easiness to use. We found a significant effect of the three layouts on ratings, $F(2, 22) = 5.73$, $p = 0.01$. Post hoc comparisons using the Bonferroni test indicated that the mean score for mirrored layout ($M = 6.08$, $SD = 0.79$) was significantly different from the projective layout ($M = 4.42$, $SD = 1.56$). However, the integrated condition ($M = 5.42$, $SD = 0.99$) did not significantly differ from the mirrored condition and projective layout. In brief, these results suggest that the mirrored layout yields better ratings of the "living room design" task (Figure 5.14(a)).

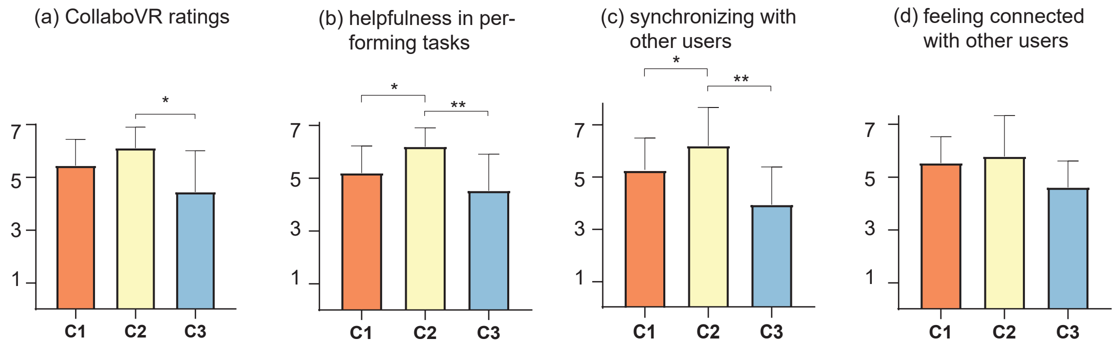


Figure 5.14: CollaboVR's ratings, degree of helpfulness users in performing tasks, in synchronizing with partners, and in connecting with partners using the integrated layout (C1), mirrored layout (C2), and projective layout (C3). *: $p < 0.05$, **: $p < 0.01$. We found a significant difference in ratings between C2 and C3; in the degree of helpfulness between C1 and C2, C2 and C3; in synchronizing with partners between C1 and C2, C2 and C3. In terms of feeling connected with partners while using CollaboVR, the statistical results differed significantly among the three conditions. However, we did not find significant differences between each pair of conditions from post hoc tests.

Additionally, we found a significant effect of the conditions on easiness to use, $F(2, 22) = 11.76, p < 0.01$. Post hoc comparisons using Holm test indicated that the mean score for projective layout ($M = 4$, $SD = 1.71$) was significantly lower than the integrated condition ($M = 6.08$, $SD = 0.79$) and mirrored condition ($M = 6$, $SD = 1.04$)(Figure 5.13(a)(b)).

A significant effect of the conditions on "helpfulness in performing tasks" was found, $F(2, 22) = 7.03$, $p = 0.004$. Post hoc comparisons indicated that the mirrored condition ($M = 6.17$, $SD = 0.72$) had significantly higher mean values than the integrated ($M = 5.17$, $SD = 1.03$) and projective layout ($M = 4.5$, $SD = 1.38$)(Figure 5.14(b)).

We also asked participants about the rankings of the layouts. 58.3% of the participants (7 out of 12) preferred the mirrored layout most, while 25% (3 out of 12) of the participants thought the integrated layout is their favorite and two participants preferred the projective layout (see Figure 5.13(b)). One sample Kolmogorov-Smirnov test indicated that the user preferences did not follow a normal distribution, $D(12) = 0.3$, $p = 0.004$ (see Figure 5.13(b)).

Those who preferred mirrored layout mentioned: *"In mirrored it is easy and convenient to communicate with others."*(P3, F). *"People didn't block my view, and I could see the content clearly."*(P5, M). *"[It is] more helpful when working on a group project. Feels like I have enough space to draw."* (P9, F).

Participants who preferred integrated layout explained that *"because it is comparable to reality."*(P2, M). P1(M) had a similar opinion *"because the real world is more similar to the integrated layout."*

Two participants preferred projective layout emphasized that *"I could sit sketching and had more control." (P7, F).* P10(M) commented: *"[it] allows drawing on the table, more intuitive to draw."*

Effectiveness for Collaboration. Task performance of each group and questions about remote collaboration were analyzed through the RM-ANOVA method. We found a significant effect of the conditions on task performance, $F(2, 4) = 98$, $p < 0.001$. Post hoc comparisons indicated that the mean score for projective layout ($M = 5.33$, $SD = 2.08$) was significantly different than the integrated ($M = 9.67$, $SD = 1.53$) and mirrored layout ($M = 9$, $SD = 2$). However, the integrated layout did not significantly differ from the mirrored layout. Therefore these results indicate that using a projective layout has a negative effect on task performance. P8(M), a designer for 3D models who frequently used the tablet for drawing, shared some feedback: *"this is like using a tablet. I preferred to spend more time on drawing the details and polishing my work when I was in this layout."*. Taking statistical results and subjective feedback into account, we think the projective layout may encourage participants to focus more on the details and better express themselves (Figure 5.13(a)(a)).

Participants thought it was easy to follow what partner was doing during the task ($M = 5.83$, $SD = 0.83$), easy to collaborate with others using CollaboVR ($M = 5.33$, $SD = 0.65$), and moderately easy to anticipate what partner planned to do next ($M = 4.92$, $SD = 1.08$). P3(F) commented that *"when [another user] started to draw the legs for the table, I quickly get his idea about the design of the legs, so he doesn't need to say what kind of legs he wants."* (Figure 5.12). Furthermore, we ran the RM-ANOVA test to compare different conditions on participants' feelings of connection and in sync with during the task. There was a significant effect of the condition on how connected do you feel to task partners, $F(2, 22) = 3.89$, $p = 0.036$. Post hoc comparisons using Bonferroni test indicated that no significant effects among three conditions: projective layout has lowest score ($M = 4.58$, $SD = 0.99$), mirrored layout has the best result ($M = 5.75$, $SD = 1.54$) and integrated layout is in the middle ($M = 5.5$, $SD = 1$) (Figure 5.14(d)).

Regarding how in-sync with the task partner during the experiment, we found a significant effect based on the RM-ANOVA results, $F(2, 22) = 9.40$, $p = 0.001$. Post hoc comparisons showed that the mean score for each condition was significantly different from each other. The mirrored layout has the best results, with $M = 6.17$ and $SD = 1.47$. The integrated condition has a better-than-neural score in average ($M = 5.25$, $SD = 1.22$), while the projective layout has an average score ($M = 3.92$, $SD = 1.44$) (Figure 5.14(c)).

Subjective Feedback. We asked participants what scenarios they would like to use CollaboVR and in which layout. The integrated layout is good for an explanation in general. P8(M) commented, *"there could be merit once you're doing something more complex."*. P2(M), who rated himself as a novice VR user, thought, *"I like integrated layout because it is very easy to understand, just like reality."*.

The mirrored layout may be the best option for presentation. P4(M) recommended it because *"you can better control your drawing, meanwhile keep an eye on people's reaction."*. P5(M) considered it from a student's perspective, *"felt like Khan Academy [Dijksman and Khan 2011] in 3D vision."*. P9(F) thought she can benefit from mirrored layout when brainstorming because no one is blocking the view, *"you can see everybody but you have your own space."*.

When discussing the suitable scenarios for the projective layout, P8(M) thought a VR live demo or presentation could be beneficial from a projective layout, especially for a time-consuming one. He described *"himself giving a presentation to other people while an audience was looking at the large monitor-like board."* and *"just want to focus on the board."*. Meanwhile, P11(M) thought it would be helpful for collaborative design and suggested us to use a pen rather than the controller.

In general, the participants found it an engaging experience and love to spend more time with friends. *"It's definitely a fun environment, entertaining."*(P7, F).

Observations. When using a mirrored layout, participants were confused about the spatial relationship in the beginning although researchers had explained it before the task. Then they quickly understood that other participants were in the "mirror". We also noticed that participants were willing to move one step aside when they were watching other participants and the content blocked the view between them and the others no matter in which condition. That suggests other alternatives should be associated with the face-to-face concept for maintaining eye contact. We also found that some of the participants preferred to look at the private workspace when in projective layout and others preferred to watch the shared board. For participants who were working on the content, providing the option to have eye contact or not for the participant is valuable.

5.9 Discussion and Conclusion

With CollaboVR, we aim to explore opportunities and challenges for creative collaboration, explore the impacts of different layouts, and better comprehend the needs and challenges for multi-user communication in VR.

Improving remote creative collaboration. We consider the effectiveness of remote collaboration from two perspectives: how CollaboVR fosters communication among participants and how CollaboVR helps collaborative work. Our user studies showed that participants felt strongly connected with task partners when using their preferred condition ($M = 6.25$, $SD = 0.86$). In addition, they managed to follow their partners' work ($M = 5.83$, $SD = 0.83$) and anticipate their partners' behavior to some extent ($M = 4.92$, $SD = 1.08$). Participants felt highly in-sync with task partners while using CollaboVR ($M = 6.33$, $SD = 0.98$). Moreover, CollaboVR was greatly helpful to users for completion of the design task ($M = 6.33$, $SD = 0.65$). We concluded that CollaboVR can foster communication and help collaboration when participants are geographically dispersed.

User preferences. We found that the mirrored layout (C2) had better usability and task performance, and received the highest ratings from the participants. Many participants mentioned that the mirrored layout helped them focus on both the content and the other participants simultaneously and that their views were not blocked because of the layout design. The integrated layout received moderate scores from participants. Participants found it to be closest to real-life scenarios. That is to say that although the integrated layout did not solve certain issues, for example, participants' arms may block the sight of the audience, participants were able to alleviate those issues as they usually did in real life while having better communication and collaboration through CollaboVR. The projective layout was rated lowest but also showed the greatest potential in detail sketching and in being a good fit for long-term work.

We envision other user scenarios for CollaboVR. For example, CollaboVR could be used to communicate with others for non-expert use such as brainstorming and presentation. Different tasks may lead to different preferences in configurations. If the collaborative task is focused on object manipulation [Salimian et al. 2019], floor plan design [Thanyadit et al. 2018], or navigation [Weissker et al. 2019; Satriadi et al. 2019], participants may want to form a circle around the object. In that case, the mirrored layout is not very effective since the focus is not on the other participants of the group but rather on the objects to be manipulated. When giving a presentation, the presenter and audience may prefer different layouts. Mirrored layout maintains the gaze between the user and the others from the user's perspective while sacrificing the gaze among others. However, the integrated layout keeps this information. Although we evaluated each layout individually, CollaboVR is a reconfigurable framework that supports real-time layout switching and easy to scale to new layouts to meet various and changing requirements.

Miscellaneous User movement and tracking capability are usually constrained within the small space around the user's desk. Even if the user is not bounded by physical space, a mirrored layout may be preferred for face-to-face collaboration; otherwise, the sketches appear reversed to the observer. Hence, the customization of user arrangements can greatly improve the overall user experience.

Projection mode leverages consistent mid-air user interaction as the direct mode. Supporting touchpads will be a nice extension for CollaboVR. However, the form factor of the current-generation touchpads may not be suitable for complex shapes.

Limitation. As a proof of concept and an example open-sourced framework, one limitation of CollaboVR is that we currently only support one application, Chalktalk. Connecting various cloud-based applications will bring more possibilities and greater capability for CollaboVR. With recent advances in neural rendering[Tewari et al. 2020], one may integrate GauGAN [Park et al. 2019], SketchCOCO [Davis 2013], and Text-based editing of talking-head [Fried et al. 2019] into CollaboVR.

Because our main contribution is the design and implementation of CollaboVR, and the exploration into different user arrangements and input modes, our user study focuses on comparison among the three layouts on a specific task, "designing a living room". A future study may enrich these results by allowing users to freely switch layouts in real-time while assigning multiple collaborative tasks for different purposes, to study how the choice of layout for a given task may affect the results.

Potential Impacts. We envision that CollaboVR will be useful for collaborative scenarios such as remote presentations and virtual conferencing. For example, web conferencing software such as Google Meet and Zoom is widely used for meetings and 2D presentations. However, it is sometimes difficult for presenters to notice who in the audience is raising hands or asking questions, while also posing a challenge for audience turn-taking. CollaboVR can help with such scenarios by providing workspace awareness. In virtual reality settings, Mozilla Hubs has been used to hold multi-user conferences with virtual avatars, yet provides very little support for creative collaborative work. CollaboVR may further extend the interaction capabilities of VR meetings by empowering participants to change meeting layouts and freely express their ideas by sketching or writing on virtual whiteboards.

We presented CollaboVR, an open-source reconfigurable framework for distributed and co-located creative collaboration in immersive environments. Our system was motivated by real-world metaphors such as side-by-side whiteboarding, face-to-face lecturing, and designing on sketchpads. We described the cloud-based system architecture, two design variables (user arrangement and input mode), rich interactive user interface, and corresponding technical details. We conducted a within-subject user study to quantitatively and qualitatively evaluate CollaboVR and compared three conditions: integrated, mirrored, and projective layouts. Our experimental results indicate that all participants can easily interact with CollaboVR and we found a significant difference in performance and ease of use in integrated layout v.s. projective layout and mirrored layout v.s. projective layout. Feedback from our interviews further suggested that CollaboVR is entertaining for communication and very helpful to foster collaboration. A few participants suggested that they would consider CollaboVR as a daily-life tool and can envision its potential for creative collaboration. Overall, the mirrored layout was mostly preferred by participants for our "design a living room" task, as it encourages more eye contact, and participants found it easy to reach a consensus when conflicts occur.

As we design CollaboVR as an extendable collaborative VR framework, we hope it will facilitate future research in collaborative work in VR, including extending the design space of sketch-based interaction, exploring effects of non-verbal cues in multi-user communication, and adding deep-learning-based models as cloud-hosted applications in CollaboVR. Eventually, virtual communication can in some ways be more effective than a physical collaboration by giving remote participants the superpower to visualize ideas with speech and sketching [Park et al. 2019], by transmitting physical or digital contents with cross-device interaction [Heun et al. 2013], and see, hear, and even feel each other by real-time reconstruction [Orts-Escolano et al. 2016] and powerful sensors [Harley et al. 2018].

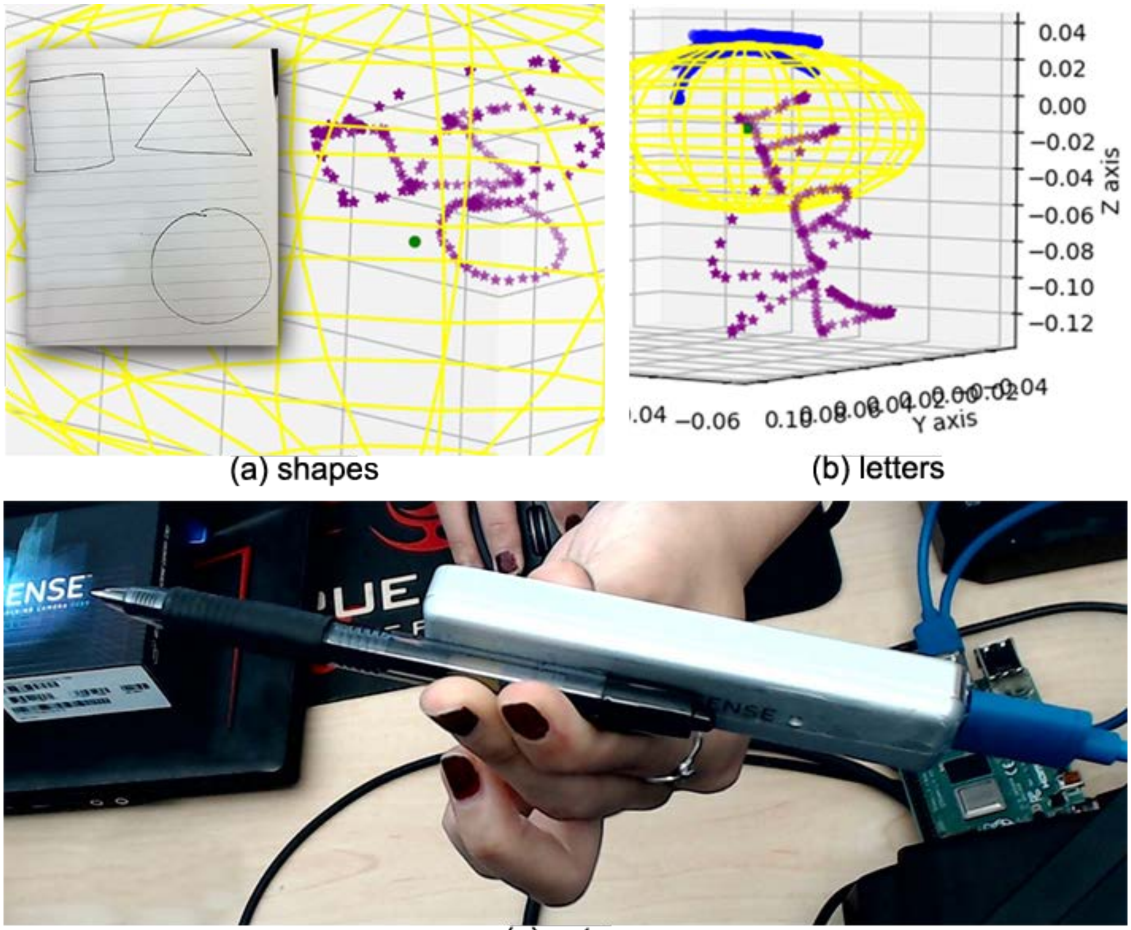
6 | Text Entry and Handwriting in VR

6.1 Summary

While playing with CollaboVR with VR controllers, I found that controllers sometimes behave as a signal that keeps reminding me my interaction is unnatural. It is one reason that users felt tired and inefficient when using VR, especially when drawing or typing letters. Besides, Virtual Reality is well known and seems to be perfect for entertainment use currently. I went back to read Ivan Sutherland's paper. "Our objective is to surround the user with displayed three-dimensional information." While VR is known for entertainment use, pushing VR for serious use, productive use is very crucial and attractive to me. Thus I did preliminary research on writing and typing via hands since text entry is one of the most frequent, important, and demanding tasks in personal computing.

6.2 Preliminary Experiment: RealSensePen and VRite

One trend I have experimented with is writing through fingers or pens. Compared to writing in the middle air, writing on a surface or a board is more convenient, natural, and easy to control. I attached a depth camera (RealSense T265) to my pen and used the reported IMU data to calculate the pentip's position. The calibration procedure is inspired by DodecaPen [Wu et al. 2017] but not the same. I managed to draw shapes and write some letters (Figure 6.1).



(a) shapes

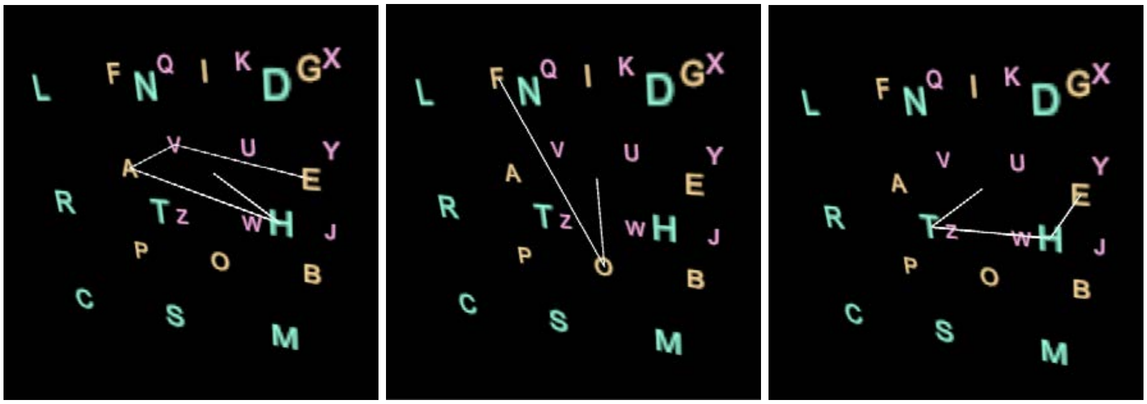
(b) letters

(c) setup

Figure 6.1: RealSensePen is composed with a regular pen and RealSense T265.

The other trend is researching virtual keyboard layout. VRite (Figure 6.2) is a glyph based system, analogous to a 3D version of SHARK [Zhai and Kristensson 2003], which powered Swype [Inc. 2016], a keyboard commonly used by mobile devices. Thus, we are confident that a trained user can achieve an acceptable word-per-minute (WPM) and accuracy. We use a glyph-recognition algorithm that allows for relatively sloppy glyphs to be accurately recognized by the system to further allow users to input glyphs at a faster rate with higher accuracy.

From the subjective feedback and quantitative result, we summarized the limitations: It was difficult to learn, as is any new keyboard system. In particular, very long words were extremely taxing to memorize and gesture out. On the other hand, short words, like the ones shown above,



(a) word "have". **(b)** word "of". **(c)** word "the".

Figure 6.2: Examples of the paths formed when writing the words "have"(a), "of" (b), and "the" (c).

were very quick to learn. It was a bit tiring to use. Moving one's hand in three dimensions gets tiring after a while, and this is a big consideration for writing systems. I believe this could be solved with some adjustments that allowed for smaller depth motions, but it would take a bit more experimentation. It didn't have a lot of features necessary for writing, like punctuation, and the rigid cube format doesn't allow for much more room. However, I'm sure I could come up with something if the above problems were solved.

6.3 Introduction

With the increasing popularity of consumer-grade head-mounted displays (HMDs) such as VR headsets and AR glasses, there is an expanding interest in text entry methods that can support such wearables [Lin et al. 2017]. For example, if smartphones are replaced by inexpensive and lightweight AR glasses, an AR analog of 'texting' will be required. It is plausible that most HMDs will allow gaze tracking and finger detection shortly to provide the best possible user experience; therefore a future text entry method will likely be able to take advantage of two faculties that have traditionally been used in typing: our fingers and our eyes. To be successful, a text entry method suitable for HMD wearers needs to address the following challenges:

Availability. HMD users are much more constrained than desktop users in what hardware they can use. Physical keyboards are often unavailable or out of reach, e.g. when users are moving freely or interacting with their real or virtual environment. In the case of AR, their environment may often be uncontrolled, e.g. on a bus, so text entry should ideally rely only on hardware that can be carried unobtrusively.

Accessibility. HMD users are constrained in their access to user interfaces. They may not be able to look at their hands or a physical keyboard, e.g. due to occlusion from a VR headset or the need to stay aware of their environment when using AR. Therefore text entry should ideally avoid the need for the user to look at hands or manual devices. Also, users may be sitting, standing, or even walking (not uncommon for mobile texting), so a text entry method should ideally be accessible from a wide range of poses.

Learnability. Because text entry is a basic task of computing systems, it should ideally be easy to learn. Because many users are already proficient in the use of a QWERTY keyboard, much previous work has aimed to exploit this familiarity to improve learnability. [Boletsis and Kongsvik 2019; Pham and Stuerzlinger 2019].

I propose TapGazer, a new text entry method designed to address these challenges. TapGazer allows users to perform text entry simply by tapping their fingers, without needing to look at their hands or be aware of finger position. Taps may be detected with any input device capable of discerning which finger is currently being tapped, e.g. finger-worn accelerometers such as TapStrap, touch-sensitive surfaces such as smart cloth, or visual finger tracking, since the location where a finger is tapped is not needed. Tracking fingers' identities and detecting whether a finger has tapped is less complicated and more accurate than tracking both the identity and location of each finger, and it is generally easier for users to focus on tapping their fingers without the need to worry about finger location. Given a suitable input device, any available surface may be used to support the hands and facilitate tapping movements, e.g. a table or one's thighs. As a result, TapGazer is designed to be beneficial not only for HMD users, but for scenarios including typing while standing, typing without a physical keyboard, and typing for mobile devices.

To enable text entry by finger tapping, TapGazer simplifies keyboard input by assigning multiple letters to each finger. Because this mapping is one-to-many, it is ambiguous (see the color-coded keyboard layout in Figure 6.3(b)). We resolve this ambiguity by showing word suggestions in the users' display and allowing them to select the correct word via gaze. As a result, users do not need to look at their fingers or input device. TapGazer's finger-to-letter mapping is based on a QWERTY keyboard layout. As a result, people can reuse their QWERTY skills and retain the performance benefits of ten-finger typing, which is generally faster than alternatives such as word-gesture keyboards [Chen et al. 2019]. TapGazer supports the entry of unknown words, symbols, and cursor navigation by allowing users to switch between different modes. Furthermore, because gaze tracking may not always be available, we describe variants of TapGazer that work without gaze tracking. I investigate the following research questions:

RQ1 How can text input be efficiently implemented using only finger taps and gaze?

RQ2 How does TapGazer perform in terms of speed, accuracy and user preference?

RQ3 How can we model user performance in TapGazer?

I address these questions by first discussing the design of TapGazer (RQ1), then reporting on a user study evaluating TapGazer (RQ2), and lastly providing a model-based analysis of how different users of TapGazer will likely perform (RQ3).

Novelty. Some previous work has looked at reduced QWERTY keyboards and word disambiguation. VType [Evans et al. 1999] applies a reduced keyboard layout, attempting to reconstruct words automatically based on finger sequence, grammar, and context, but does not allow users to choose between ambiguous words. The 1Line keyboard [Li et al. 2011] and the stick keyboard [Green et al. 2004] flatten the QWERTY keyboard from three rows to one, allowing users to choose between ambiguous words through touchscreen gestures and arrow keys. Yet to the best of our knowledge we are the first to investigate tapping while resolving ambiguity through the gaze. Also, the performance we measured for TapGazer (average 52.5 wpm with gaze tracking no completion, max. 81.3 wpm) surpasses that reported for similar works (see Table 6.1).

6.4 Related Work

A key requirement of manual typing approaches – typing with hands – is detection and tracking of hands or fingers. Various methods for this have evolved over the last decades, including using gloves [Lee et al. 2002; Thomas and Piekarski 2002], markers [Han et al. 2018; Markussen et al. 2014], audio signals [Wang et al. 2014], cameras [Yin et al. 2016], and specific devices such as Leap Motion [Yi et al. 2015]. Moreover, various input recognition methods have been proposed, with some recognizing input as single characters ('character-level') and others recognizing entire words ('word-level'). Methods recognizing larger chunks of input (e.g. words, sentences) are typically more effective than those recognizing characters [Vertanen et al. 2018]. Lastly, there are good online input prediction and correction methods that can be used to improve the performance of text entry [Goodman et al. 2002; Zhang et al. 2019; Oulasvirta et al. 2013]. To develop a fast and usable text entry design using tap and gaze, we closely investigated prior work in alternative keyboard layout design across HMD and non-HMD scenarios, gaze interaction, and text entry in VR/AR. An overview of the most relevant and fastest methods, with their average speeds listed in words per minute (WPM), is shown in Table 6.1.

6.4.1 Alternative Keyboard Layouts

Alternative layouts generally aim to increase performance, often while supporting a limited interaction size with a reduced number of keys, which makes them relevant for TapGazer. Many layouts are designed based on optimization of typing performance, e.g. metropolis keyboard [Zhai et al. 2000] and GK-D and GK-T [Smith et al. 2015]. Another common consideration is the similarity to familiar layouts such as QWERTY or T9 for learnability, e.g. for mobile phones [Dunlop et al. 2012; MacKenzie et al. 2001], smart glasses [Ahn and Lee 2019], and smartwatches [Qin et al. 2018]. Familiar layouts are often adapted to new typing gestures, e.g. using thumb-to-finger interaction for small-screen devices or VR/AR using split QWERTY [Olofsson 2017; Whitmire et al. 2017] or T9 layouts [Wong et al. 2018]. Another trend is rearranging keyboard characters into different 2D or 3D shapes: QuikWriting [Perlin 1998] and its gaze-version [Bee and André 2008] distribute letters into a circle; PizzaText [Yu et al. 2018], WrisText [Gong et al. 2018], and HiPad [Jiang and Weng 2020] use a pie-shaped layout; Keycube [Brun et al. 2019] attaches push buttons to a physical magic cube for typing.

When applying a reduced keyboard layout, fingers or keys are not uniquely assigned to characters, so a mechanism for disambiguation becomes necessary. LetterWise [MacKenzie et al. 2001] uses prefix-based rather than word-based disambiguation, i.e. users press a button if the current character is wrong and then the respective character of the next-likely prefix is shown. By repeatedly pressing the button, even non-dictionary words can be typed. Stick keyboard [Green et al. 2004] compresses the QWERTY keyboard into one line, with each key mapped to 2-3 characters. Users choose one of several ambiguous words by scrolling through possible candidates with button presses. Similarly, 1Line keyboard [Li et al. 2011] reorganizes the QWERTY keyboard to a single line specifically for touchscreen typing, using touch gestures to support candidate selection. Similar to these works, TapGazer is based on a reduced QWERTY layout, but it uses different mechanisms for faster disambiguation.

6.4.2 Gaze-assisted Text Entry

Text entry with gaze does not require a physical keyboard, therefore it is a natural option to consider for HMDs, which can incorporate gaze trackers. Gaze-only methods mainly fall into the following categories [Majaranta and Räihä 2007]: direct gaze pointing with dwell (“gaze typing”), eye switches, discrete gaze gestures, and continuous gaze gestures (“gaze writing”). Dwell [Benligiray et al. 2019; Huckauf and Urbina 2008; Majaranta et al. 2009] (i.e. looking at keys for a certain time to trigger clicks) has been widely applied and optimized to solve the Midas Touch problem [Jacob 1993] (i.e. inadvertent clicks). Approaches for reducing the dwell time necessary for each key have been explored, e.g. by dynamically adjusting it based on prefix [MacKenzie and Zhang 2008], word frequency, or character placement [Mott et al. 2017]; however, it is still a major factor slowing down typing speed. Eye-switch approaches try to avoid dwell by using other operations such as blinking, brow interaction, and head movements [Gizatdinova et al. 2012] as triggers. Similarly, discrete gaze gestures have been proposed to avoid dwell, e.g. by adding a resting zone in the typing area [Bee and André 2008], 'swiping' over a keyboard with gaze to enter a word [Kurauchi et al. 2016; Cheat and Wongsaisuwan 2018], or using other confirmatory eye movements such as inside-outside-inside saccades [Sarcar et al. 2013]. Some disambiguation algorithms have been proposed to improve the accuracy of word-level gaze gestures [Liu et al. 2015; Pedrosa et al. 2015]. Dasher [Ward et al. 2000] uses continuous gaze gestures to zoom towards and select candidate letters and words. Gaze-only text entry methods are much slower than typing with a keyboard; therefore some approaches try to speed it up by using other modalities for key and word selection, e.g. a physical button [Kumar et al. 2007], a brain-computer interface [Ma et al. 2018], or touch gestures [Kumar et al. 2020; Ahn and Lee 2019]. If gaze tracking is not available, many gaze-based approaches can be modified to use head movement only [Yu et al. 2017; Xu et al. 2019b]. This can be combined with other head gestures, e.g. nodding for letter selection [Lu et al. 2019]. Overall, gaze-based text entry methods facilitate social privacy and can be used while standing or moving in VR [Rajanna and Hansen 2018]; however, they are still much slower than physical keyboards (below 25 WPM) as gaze movements that require fixations between saccades are generally time-consuming (in the order of 100-400 ms [Findlay 1997]). Therefore, TapGazer uses gaze for disambiguation rather than typing.

6.4.3 Text Entry in VR/AR

Various methods have been investigated for text entry in VR/AR [Dube and Arif 2019]. Because text entry using a physical keyboard is faster than other typing solutions, many approaches for text entry in VR/AR try to facilitate access to a standard physical keyboard rather than replace it. This has mainly been done by tracking and visualizing a physical keyboard in VR while sitting at a desk, either by blending in a video stream showing the real keyboard [Jiang et al. 2018; Lin et al. 2017; McGill et al. 2015; Bovet et al. 2018] or by visualizing the keyboard in VR [Knierim et al. 2018; Grubert et al. 2018; Walker et al. 2017; Bovet et al. 2018; Otte et al. 2019]. To support better mobility, HawKEY [Pham and Stuerzlinger 2019] uses a portable keyboard for users to type on while standing and walking in VR. These approaches show that using a physical keyboard and high-quality tracking leads to good performance, although the setup can be complex. However, most HMD users are unlikely to carry a full-sized physical keyboard around with them, and handling such a keyboard while moving and interacting in an uncontrolled environment can be cumbersome.

Other work has investigated VR/AR text entry with pointing gestures on virtual keyboards. Xu et al. [Xu et al. 2019a] and Speicher et al. [Speicher et al. 2018] compared pointing methods to selecting virtual keys with controllers, head and hand. Boletsis & Kongsvik [Boletsis and Kongsvik 2019] proposed virtual keyboard layouts to optimize controller-based key selection. PizzaText [Yu et al. 2018] arranges virtual keys in a circle separated into segments. Anonymous [Surname 2018] compared controller-based with stylus-based virtual keyboard interaction. Yanagihara et al. [Yanagihara et al. 2019] introduced a curved virtual QWERTY keyboard, allowing users to use a controller to swipe between different keys (21 wpm). Similarly, Chen et al. [Chen et al. 2019] proposed word gestures by pointing and swiping at a virtual keyboard. Additionally, Dube and Arif [Dube and Arif 2020] researched the impact of key design on virtual keyboards for typing speed and accuracy. While these approaches improve mobility, they are much slower than physical keyboards, with typical speeds below 25 WPM.

Many VR/AR text entry methods use fingers or hands directly. A popular approach is to detect pinch gestures between fingers and thumbs, e.g. using a data glove. Pinch keyboard [Bowman et al. 2001] combines pinch with hand rotation and position to select letters. KITTY [Kuester et al. 2005] uses pinch gestures on different parts of the thumb. PinchType uses a reduced keyboard, and if necessary, allows the user to disambiguate words with hand gestures [Fashimpaur et al. 2020]. DigiTouch [Whitmire et al. 2017] uses continuous touch position and pressure. Quadmetric [Lee et al. 2019] and HiFinger [Jiang et al. 2019] support one-handed text entry with pinch. BlindType [Lu et al. 2017] uses single-thumb touchpad gestures. RotoSwype [Gupta et al. 2019] uses one-handed word gestures by rotating a ring worn on one hand. Yu et al. propose one-dimensional 'handwriting' of words with a tracked finger or controller [Yu et al. 2016]. Pinch and word gesture-based approaches are flexible but slow, with typical speeds far below 20 WPM. Also, mid-air finger gestures can be hard to track and can lead to fatigue when performing longer tasks [Adhikary 2018; Dudley et al. 2018]. FaceTouch [Gugenheimer et al. 2016] allows users to type on a touch surface attached to their HMD. ARKB [Lee et al. 2003] proposes visual tracking of fingers for tapping on a virtual QWERTY keyboard. VISAR [Dudley et al. 2018] facilitates mid-air one-finger tapping on an AR QWERTY keyboard. VType [Evans et al. 1999] uses finger tapping on a reduced QWERTY keyboard layout and reconstructs words based on finger sequence, grammar, and context for text input in VR. The accuracy reported for a predefined vocabulary is high; however, no method for disambiguation between candidate words was considered and no typing speed was reported. Tapping on a reduced QWERTY keyboard is promising for text entry in VR/AR as it is flexible and robust compared to alternatives. Therefore, we explore how it can be optimized by using gaze input and additional taps for disambiguation.

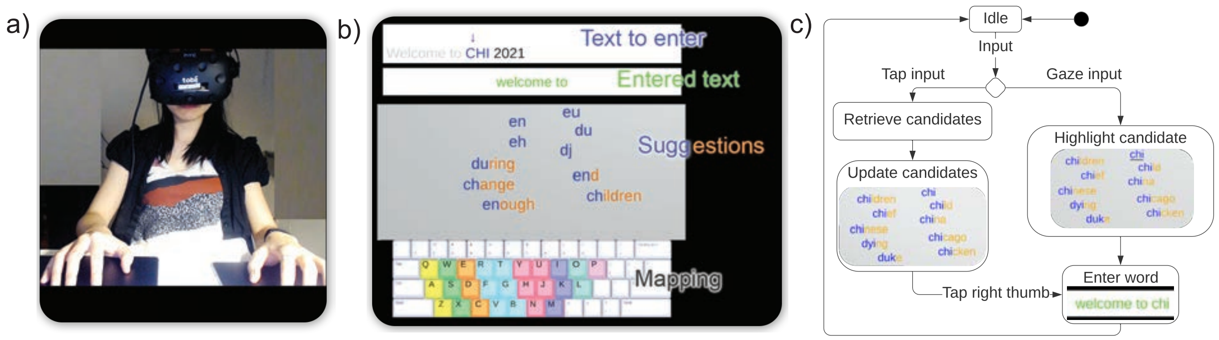


Figure 6.3: a) Physical setup: The user enters text without needing to see hands or keyboard, by tapping on a surface and resolving ambiguity between candidate words via gaze selection. b) Visual interface: Fingers are mapped to multiple letters (see colors at the bottom); the central area shows candidate words corresponding to the current input sequence of finger taps. Users can select a word by gazing at it and tapping the right thumb. c) State machine of TapGazer with gaze selection and word completion.

6.5 Design and Implementation

TapGazer allows users to tap words as if they are typing them on a physical QWERTY keyboard, and then disambiguate their tap input selecting the desired word through the gaze. It was designed primarily for HMD users, but could also be useful for other scenarios where more conventional input devices are unavailable or difficult to access. Given suitable sensors, users can type by tapping their fingers on any surface or even in mid-air. As TapGazer only considers the identity of the finger that is currently tapping and not its position, it only needs to know which of the user's 10 fingers has just been tapped, if any, at any given time. Each of the 26 letters of the alphabet is mapped to one of the eight non-thumb fingers, while the two thumbs are reserved for controlling use including word selection, deletion, etc. Figure 6.3 illustrates the state machine of TapGazer with gaze selection and word completion. Starting from an idle state, TapGazer waits for tap or gaze input events. Except for the thumbs, a finger tap adds a letter to the *input string*, starting from an empty string. The input string is constructed from an *input alphabet* with one character for each of the eight fingers: we are using the characters `asdfjkl;`, which correspond to the rest positions of each finger on a QWERTY keyboard, for later reference. When typing a word with TapGazer, the user taps the fingers as they would do when typing on a QWERTY keyboard. However, as each finger tap can be interpreted as one of several characters, the word represented by the input is ambiguous: for example, `fjd` is the input string for the words ‘the’ and ‘bye’. We refer to a set of words that all have the same tapping input string as a *homograph set*. A tap with the left thumb deletes the current input string so users can start the word again. A tap of the right thumb selects the word to enter from a list of suggestions while the word is pointed at by the user’s gaze.

As a user enters an input string, the central area of TapGazer’s user interface shows a list of word *candidates*: similar to predictive text on a mobile phone, the user is given a list of the most likely words to choose from. TapGazer shows all words in the *homograph set* for the given input string, which we call *completes candidates* as they are based on the whole input string (e.g. ‘the’ for `fjd`). Additionally, TapGazer uses a language model to show the most likely *incomplete candidates*, i.e. words with a prefix matching the current input string (e.g. ‘these’ for `fjd`). After each tap, TapGazer updates the candidates shown. To select a candidate, the user looks at it, and in response, the fixated candidate is highlighted with an underline. If the right thumb is tapped, the currently highlighted candidate is selected and added to the entered text. At this point, the TapGazer state machine starts again with an empty input string. If the user taps the right thumb but does not fixate any candidate, then the most likely candidate is selected based on a language model. Figure 6.4 illustrates how to type ‘chi’ with TapGazer. Word completion in TapGazer can be disabled; in this variant, only complete candidates are shown if they exist. If no complete candidate exists, we show the shortest incomplete candidate to inform users about the progress of typing. Furthermore, we have designed a purely manual variant of TapGazer without gaze tracking, allowing users to disambiguate candidates with extra taps. Figure 6.6 illustrates different input devices and variants of Tapgazer.

In answering RQ1, several design decisions were made: First, we use finger tapping so that users can ‘type’ on any surface and require no context knowledge between the surface location

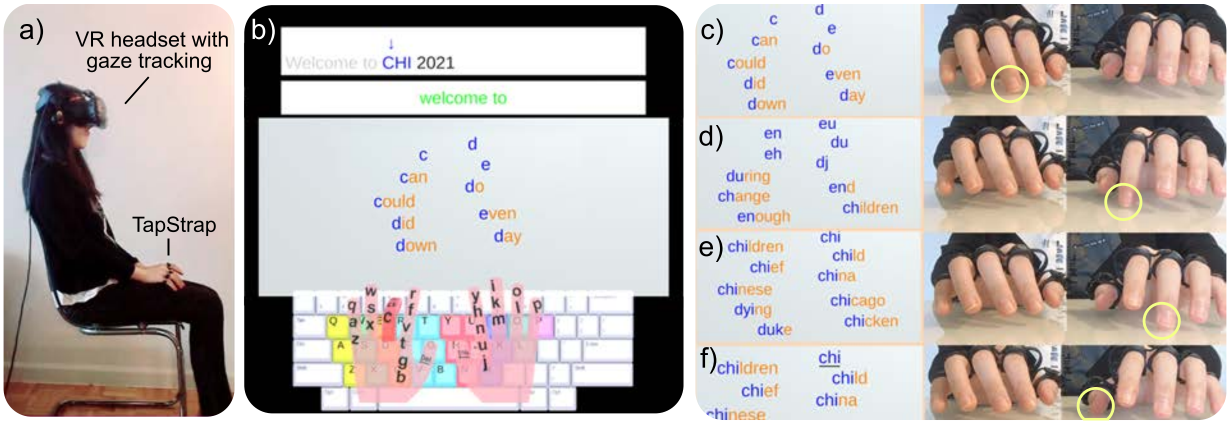


Figure 6.4: Text entry example: a) A user is 'typing' on her thighs using a TapStrap device instead of a keyboard. b) The user just started to 'type' the word "CHI". The interface provides optional visualizations of the finger-key mapping as a virtual keyboard and/or hands. c) The user first tapped the left middle finger, which is mapped to 'c', then d) the right index finger, and e) the right middle finger. f) Finally, the user looks at the word "chi" (underlined because of being gazed) and taps the right thumb to select the highlighted word.

and finger/hand location. Second, we help users find the word to type in the list of candidates, by facilitating visual search in the layout of the graphical interface. Third, we provided word completion and compare whether word completion benefits TapGazer in terms of performance.

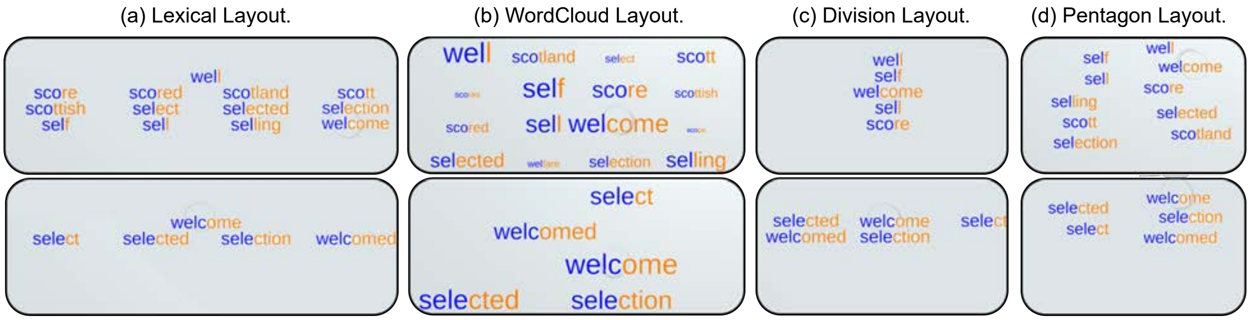
6.5.1 Virtual Keyboard Layout

Customization. TapGazer reuses the standard QWERTY layout to avoid an extra learning curve from new keyboard layouts. However, in our pilot studies, we found people have varying finger preferences for typing on the QWERTY keyboard, e.g. key 'b' may be pressed with either the left index finger or the right index finger. As a result, TapGazer creates a profile for each user to record their finger-to-key mapping. To guide novice users, we optionally visualize the customized finger-to-key mapping in a virtual keyboard and/or a hand model (Figure 6.4b), with each key colored according to its associated finger and letters rendered on their corresponding fingers. We use prefix trees to quickly lookup complete and incomplete candidate words and their word frequencies for each input string, which are generated based on the users' mapping.

Feasibility. Text entry is only feasible if all the words in the homograph set of any input string can be somehow selected. The *minimum candidate number* (MCN) is the minimum number of candidate words the interface must be able to disambiguate at a time. It is equal to the maximum number of homographs an input string can have, i.e. it describes the worst possible ambiguity that may need to be resolved. The design needs to determine the MCN in advance because display space needs to be adequately allocated, or users must be given the option to page through sets of candidates. The MCN is also relevant for performance as it describes the worst-case scenario of visual search for the right candidate. We determined popular QWERTY-based finger-to-key mappings in pilot experiments and then simulated to determine their overall MCN based on different word sources: the 1000 most common words retrieved from Wikipedia with $\text{MCN}_{\text{1K}} = 4$; the standard MacKenzie phrase corpus [MacKenzie and Soukoreff 2003], which contains 500 phrases for evaluation use, with $\text{MCN}_{\text{MacKenzie}} = 6$; and the 90% most frequent words (7,440) generated from the wordfreq library [Speer et al. 2018], which includes many very-low-frequency specialized words and acronyms that are not typically part of dictionaries, with $\text{MCN}_{\text{7K}} = 7$. We design our interface to be able to show at least 10 candidates in order to cover all English dictionary words and also many low-frequency non-dictionary words across typical QWERTY finger-to-key mappings. For unsupported words such as neologisms and special acronyms, we provide a *spelling mode* for character-level text entry (see subsection 6.5.4).

Alternative Layouts. We also calculated the MCNs of standard keyboard layouts other than QWERTY to gauge their suitability for use in TapGazer. Optimal word gesture keyboards such as GK-D ($\text{MCN}_{\text{MacKenzie}} = 11$, $\text{MCN}_{\text{7K}} = 12$) and GK-T ($\text{MCN}_{\text{MacKenzie}} = 7$, $\text{MCN}_{\text{7K}} = 17$) [Smith et al. 2015] have higher MCNs, probably because they are not optimized for key-based typing. If the left thumb is used for tapping instead of deletion (e.g. by triggering deletion with a chord), having 9 fingers to tap reduces ambiguity in the finger-to-key assignment, potentially decreasing the MCN. We calculated the MCN for some known 9 finger layouts: standard T9 [Wong et al. 2018] ($\text{MCN}_{\text{MacKenzie}} = \text{MCN}_{\text{7K}} = 5$); HiFinger [Jiang et al. 2019], which distributes letters in lexical order over nine keys ($\text{MCN}_{\text{MacKenzie}} = 5$, $\text{MCN}_{7\text{K}} = 8$); and the quadmetric optimized layout [Lee et al. 2019] ($\text{MCN}_{\text{MacKenzie}} = \text{MCN}_{7\text{K}} = 4$). Finally, we performed an extensive combinatorial search of non-QWERTY layouts and found that there is a very large number of mappings for eight fingers with $\text{MCN}_{\text{MacKenzie}} = \text{MCN}_{7\text{K}} = 4$. These results suggest that layout optimization can help to reduce the number of candidates that have to be shown at one time, which could speed up text input.

6.5.2 Word Candidate Layout


(a) Lexical Layout.

(b) WordCloud Layout.

(c) Division Layout.

(d) Pentagon Layout.

Figure 6.5: Evolution of TapGazer layout designs: a) *Lexical Layout* places the most common candidate word in the first row and arranges the other candidates in alphabetical order. All the candidates have the same font size. b) *WordCloud Layout* emphasizes frequent candidates with a larger font size. Candidates that were already shown on the previous tap keep their position. c) *Division Layout* divides all candidates into three columns according to their last letter. d) *Pentagon Layout* orders the candidates based on frequency and arranges the candidates in a compact single or double pentagon shape.

The most important part of TapGazer's visual interface is the central gray area where word candidates are shown for selection by the user (Figure 6.4b). These candidates are colored to indicate the tapping progress of each word: the prefix of each word that has already been tapped is colored in blue, while yet-to-be-tapped postfixes are colored in orange. Complete candidates are completely blue and are always shown in the interface as they must be available as word choices. Any further available space can be filled with incomplete candidates, indicating options for word completion. The number of candidates shown is a trade-off between saving taps through word completion, and visual search time spent looking for the right candidate. Visual search time is affected by the way we arrange the candidates, therefore we designed, tested, and re-designed the layout to reach a suitable design. Figure 6.5 illustrates the design evolution of TapGazer's candidate layout.

Initial Design. We first designed (a) Lexical Layout and (b) WordCloud Layout based on the following design principles. *Systematic locations*: Users should intuitively know where to look for a word. *Salience*: More likely words should be more salient (e.g. larger or more central). *Continuity*: Avoiding changes in the position of a suggested word between taps may help users to spot it. Lexical Layout places the most frequent word into the first line and the other candidates in alphabetical order below. This prioritizes systematic locations over continuity, as candidates' positions may change between taps, e.g. "welcome". WordCloud Layout arranges candidates in word-cloud style, with more frequent words arranged at the center and in a larger font. Candidates keep their positions between taps, prioritising continuity over systematic locations. To understand the effects of the layouts and their design principles on novices, we conducted a formative study with 12 participants (5 female, 7 male; aged 18 to 30, $M = 24.67$, $SD = 3.94$), comparing the two layouts in a within-participant design. Each participant used each layout twice for 5 minutes each time, followed by quantitative and qualitative questionnaires collecting their feedback on each layout and design principle. The quantitative results showed that there were no significant differences in typing speed in WPM ($F(1, 11) = .81, p = .39$) and scale of usability scores (SUS) [Bangor et al. 2008] ($F(1, 11) = 2.1, p = .15$), but Lexical layout achieved better accuracy TER% [Soukoreff and MacKenzie 2003] ($F(1, 11) = 7.1, p = .004^{**}$) and lower NASA-TLX task load scores [Hart 2006] ($F(1, 11) = 14.32, p < .001^{***}$). Participants were split half-half in their layout preference. They immediately understood the Lexical Layout's systematic locations but did not find them very helpful. Having the most frequent word at the top or center was found useful, but variations in font size were found to be distracting when typing low-frequency words. Some participants noticed WordCloud's continuity but did not find it very helpful as tapping is too fast to visually follow candidates. This was an important lesson: It is not useful to design the visual layout around the tapping process, as fingers are much faster than the eye. It is more useful to consider the layout as a pure visual search task, where visual search time is correlated with the number of candidates and the distance of eye movement [Ohkita et al. 2014].

Final Designs. Division Layout (Figure 6.5c) distributes candidates into three columns according to their last letter, ordering each column by word frequency. The column boundaries were chosen to balance the expected number of candidates in each column, with words ending in A-E on the left, F-R in the middle, and S-Z on the right. This layout is designed for power users who have learned where to expect a word, potentially reducing search time by 2/3. Pentagon Layout (d) is designed to be suitable also for novices. It arranges candidates in compact groups of five, close together to minimize eye movement but with enough separation for accurate gaze selection (at least 0.5° visual angle). The pentagon shapes mitigate overlap between long adjacent words and try to take advantage of people's ability to quickly scan groups of five items at a time [McElree and Carrasco 1999]. Complete candidates are always shown before incomplete candidates, with frequent words closer to the top. We chose Pentagon Layout for our main study as it is easier to use for non-experts.

6.5.3 Disambiguation

After presenting possible word candidates, users need to select a candidate to disambiguate the input. In text entry on mobile devices, word candidates are commonly selected by touch, and users typically first look at the candidate they want to select, which is a subliminal step. TapGazer takes advantage of this by employing gaze tracking for word selection to minimize taps and reduce cognitive load. Once the user has found the right word, she can select it with a tap of the right thumb. We chose to use a tap rather than a gaze-dwell for selection as the latter is much more time consuming and can lead to Midas Touch (inadvertent activations) [Penkar et al. 2012].

In the absence of gaze tracking, we provide a variant of TapGazer with purely *manual selection* (Figure 6.6 bottom-right). In this variant, selecting a candidate is a two-step operation: 1) tapping

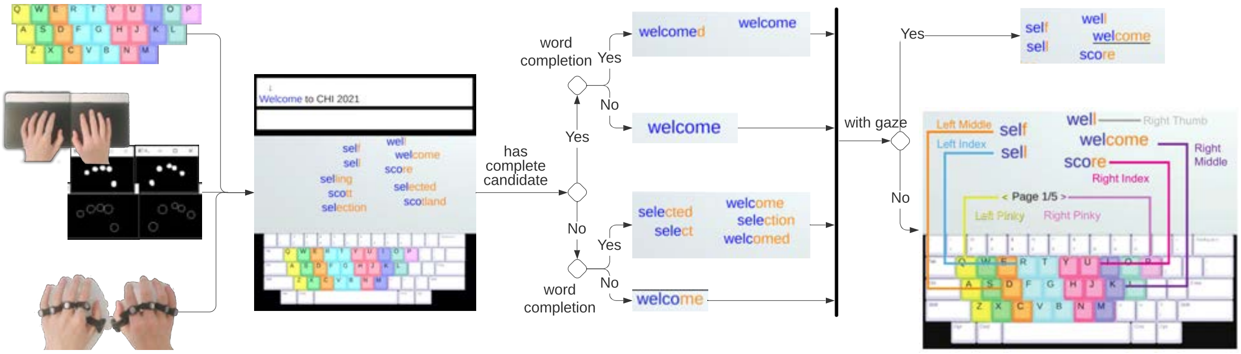


Figure 6.6: TapGazer's workflow: After receiving a tap from a suitable input device, TapGazer updates the candidates according to the word completion mode, and allows users to select a candidate either with gaze or with additional taps.

with the right thumb, and 2) tapping with one of five fingers (right thumb, right middle, right index, left middle, left index) to select one among a maximum of five candidates shown. To support selection from more than five candidates, users can page through sets of five candidates with their left and right little fingers. The layout design helps to avoid paging operations by showing complete candidates first and ordering complete and incomplete candidates by their descending word frequencies.

6.5.4 Miscellaneous Text Entry Functionality

We have designed miscellaneous text editing functions for TapGazer to make it a complete text entry method. *Deletion* of the current input string is performed by tapping the left thumb, allowing users to start a word again. If the left thumb is pressed right after selecting a candidate, the candidates for the last input string will be shown again, allowing users to change the selection or tap the left thumb again to delete the word. *Spelling mode* is triggered with a chord operation. Users can switch between word-level and character-level text entry by tapping their left and right index fingers simultaneously. Afterward, users can rotate through the characters mapped to each finger by repeatedly tapping a respective finger, and enter the character by tapping the right thumb. Tapping the right thumb again concludes the character-level input. *Cursor* *navigation* with gaze is performed by selecting words in the entered text directly with gaze and right thumb [Sindhwani et al. 2019], or by entering a cursor navigation mode through a button in the periphery of the interface [Lutteroth et al. 2015] with gaze and right thumb. Users can then move the cursor by tapping the left/right index finger and exit cursor mode with a right thumb tap. If the gaze is unavailable, users can enter cursor mode by tapping the right index and ring fingers simultaneously.

6.5.5 Design Limitations

We observed that some QWERTY users occasionally use different fingers for the same key. TapGazer currently does not support this behavior as finger-to-key mappings only allow keys to be mapped to a single finger. Many-to-many mappings are in principle possible, but would likely increase the MCN and the visual search time. Word prediction is currently based only on word frequency; it could be substituted by a more advanced method taking also the context of a word into account. We currently do not provide auto-correction, as this could confound our study of accuracy.

6.5.6 Implementation

TapGazer is implemented via Unity, Python, and C++. Figure 6.6 illustrates the workflow of TapGazer. TapGazer took finger tap as input. The input string is composed of finger IDs. Currently, we support using TapGazer with a regular keyboard, touchpad, and wearable devices such as Tap Strap. The candidates are updated based on the input string. In *without word completion* mode, we only show complete candidates when exists, otherwise we show the shortest incomplete candidate to inform users of the progress of typing. When enabling word completion, we show complete candidates first and followed by incomplete candidates order by frequency. Lastly, users can select the word by fixating the word and tap with the right thumb with gaze or tap with thumb and another finger without gaze.

Versatile input

As how TapGazer is designed, TapGazer took finger IDs as input. We provided implementation for three input devices. 1) Keyboard. The keyboard is partitioned into different areas. Each area is mapped to one finger. TapGazer takes the finger ID for the next step. The partition method is consistent with the user's finger-to-key mapping so the user can retain their QWERTY skills. 2) Touchpad. We are using sensor morph as an example. Such touchpad reports pressure image every frame. We detect the hand direction (left hand or right hand) and finger identification based on the shape of the pressure and temporal information. The detection algorithms are written in C++ and communicate with TapGazer through TCP. The formative study shows the accuracy of the finger identification detection on touchpad reached 99.86%. 3) Wearable devices. A wearable device like Tap Strap reports tapping information through Bluetooth. Hence TapGazer uses the tap information directly.

Personalized finger-to-key mapping

Personalized keyboard layouts are generated via Python. We used the Python library wordfreq [Speer et al. 2018] as our source for word frequency. We generated a word list that covers the 90% most frequent words from wordfreq as our dictionary. For each user, a profile is generated according to that users' preferred finger-to-key mapping. All the words in the corpus were iterated to generate a tap tree and a complete-word tree. For each word, we iterated the word character by character, found the corresponding finger for each character, and then created a mapping from the finger sequence. For example, the word "this" was added to finger sequence `f` , `fj`, `fjk`, and `fjks` in the tap tree, and also added to finger sequence `fjks` in the complete-word tree. All the items for any given finger sequence are then sorted based on word frequency.

Gaze selection

Gaze selection implementation is associated with the gaze tracker. We support Tobii devkit for HMD users and Tobii tracker bar for non-HMD users. Unity SDK provided by Tobii reports the coordinates of the user's gaze. A small circle is rendered as the gaze indicator so users know how the system perceives the gaze information. Based on the coordinates we can determine which candidate word is being gazed at or which area users are looking at.

6.6 Simulated Study

We conducted a user study to examine how the different TapGazer variants perform in terms of speed, accuracy, and user preference (RQ2) as compared to a physical QWERTY keyboard. We used a within-subject design with five conditions: standard QWERTY keyboard (K), TapGazer with gaze candidate selection and word completion (GC), TapGazer with gaze candidate selection and no word completion (GN), TapGazer with manual candidate selection and word completion (MC), and TapGazer with manual candidate selection and no word completion (MN). The order of the five conditions was counterbalanced to mitigate the effects of learning and fatigue.

The study had to be conducted remotely using videotelephony. Participants were using their personal computers running Windows or macOS, so participants were tapping on their keyboards when using TapGazer. Participants were free to decide where on the keyboard they wanted to tap each finger, i.e. which fingers would tap which keys. We helped them to assign keys along with the natural ranges of motion of each of their fingers so that they could comfortably tap their fingers without having to consider the keys. Equally, TapGazer did not consider individual keys but only used the keyboard to identify which finger was tapped. As most participants did not have access to gaze trackers, we impressed on them to locate the right candidates with their eyes as a simulation of using a gaze tracker in GN and GC. If the right word was shown in the candidate area, we assumed that participants selected the correct word when tapping with the thumb. If the right candidate was not shown, this would result in an incorrect word; so in line with real gaze racking, participants had to first spot the right candidate to ensure a correct input was made.

Measures. We compared the conditions using objective measures [Soukoreff and MacKenzie

2003] such as typing speed (WPM), total error rate (TER), and average times of frequent operations, as well as subjective measures such as SUS usability scores [Bangor et al. 2008] and NASA-TLX task load scores [Hart 2006]. Each tap/keystroke operation was recorded for analysis.

Procedure. After a brief introduction of TapGazer, participants gave consent to join a video call and share their screen during the experiment. First, we measured their QWERTY typing behavior and generated their customized virtual keyboard for TapGazer. Then they performed each of the five conditions (K, GC, GN, MC, MN) in counterbalanced order, with each condition consisting of a training session and five test sessions. In the training session, participants received brief instruction on the use of the respective text entry method and were able to practice it for a couple of minutes. In the five test sessions, participants were asked to enter phrases randomly sampled from the MacKenzie & Soukoreff corpus [MacKenzie and Soukoreff 2003], as fast and accurately as possible while looking at the screen. Each test session was one minute long and participants were allowed to take short breaks between the sessions. After each condition, participants completed SUS and NASA-TLX questionnaires. Each condition took around 10-15 minutes. After finishing all five conditions, participants completed a demographics questionnaire and ranked the conditions and their selection and completion modes according to their preference. Lastly, participants were interviewed about TapGazer, the reasons for the rankings, and their general suggestions. On average, the experiment took about 60 to 90 minutes.

Participants. We recruited 20 participants (3 female and 17 male; aged 24 to 35, $M = 28.25$, $SD = 3.45$) via word of mouth and flyers. 15 had used eye-tracking devices before. The average typing speed was 67.13 WPM with a TER of 8.47% ($SD = 0.025$) on a standard QWERTY keyboard.

6.7 Results

We validated that the data satisfies the assumptions of repeated-measures analysis of variance (ANOVA). We used one-way repeated measures ANOVAs to compare effects across all five conditions and their sessions, and two-way repeated-measures ANOVAs to compare the effects of the different TapGazer variants with regards to the factors Gaze (gaze vs. manual) and Completion (word completion vs. no word completion). Paired t-tests with Holm correction were used for all pairwise comparisons between conditions. All tests for significance were made at the $\alpha = 0.05$ level. The error bars in the graphs show the 95% confidence intervals of the means.

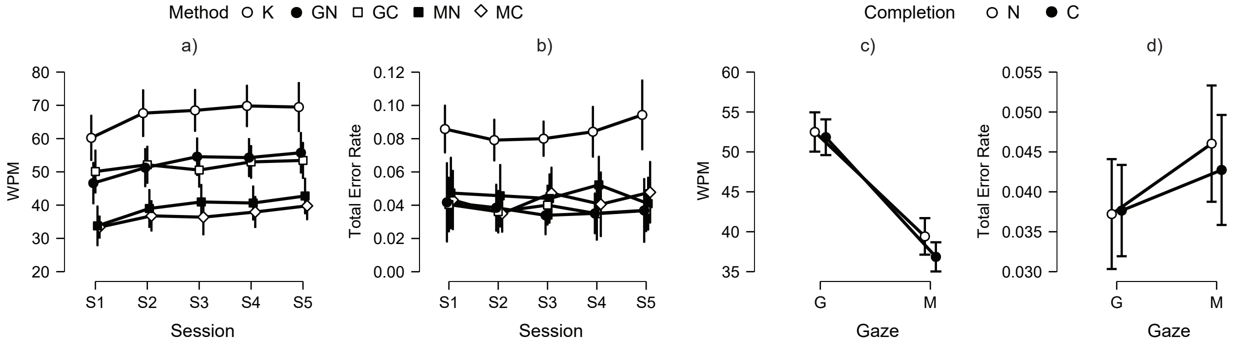


Figure 6.7: Evaluation results for WPM and TER comparing QWERTY keyboard (K) and TapGazer in GN, GC, MN, and MC modes.

Text Entry Speed. Figure 6.7a shows the average text entry rate for the five conditions. Users typed at $M = 52.49$ ($SD = 13.44$) for GC, $M = 51.84$ ($SD = 8.92$) for GN, $M = 39.42$ ($SD = 10.75$) for MC, and $M = 36.85$ ($SD = 8.47$) for MN. The main effects of Condition ($F(4, 76) = 69.56, p = .001^{***}$) and Session ($F(4, 76) = 19.81, p < .001^{***}$) were both significant. K was significantly faster than all TapGazer variants ($t(18) \geq 7.13, p < .001^{***}$). For TapGazer (Figure 6.7c), the main effect of Gaze ($F(1, 19) = 74.98, p < .001^{***}$) was significant with a 'large' effect size ($\omega^2 = 0.37$), indicating that gaze selection was faster. The main effect of Completion ($F(1, 19) = 1.517, p = .23$) and the interaction effect ($F(1, 19) = 1.92, p = .18$) were not significant.

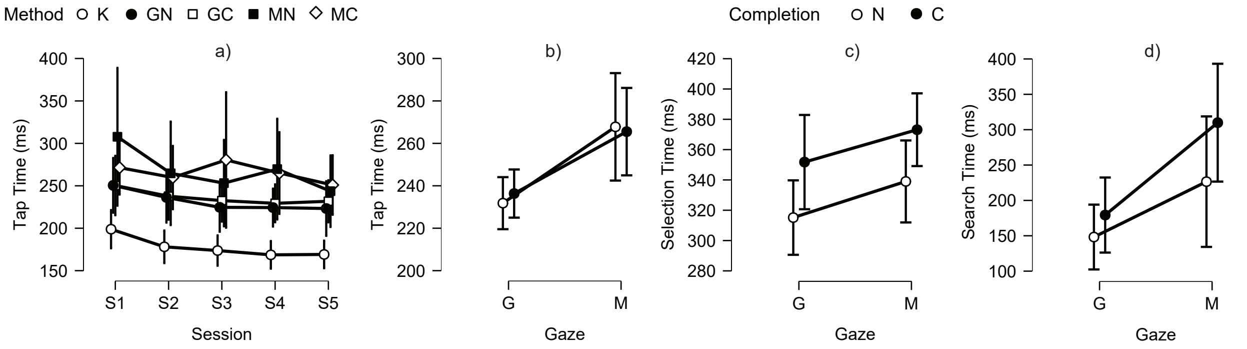


Figure 6.8: Evaluation results for operation times comparing QWERTY keyboard (K) with TapGazer in GN, GC, MN, and MC modes.

Tap Time. Figure 6.8a shows the average times per tap when entering letters of a word. The main effects of Condition ($F(1, 19) = 13.71, p < .001^{***}$) and Session ($F(1, 19) = 7.11, p < .001^{***}$) were both significant. Tapping with K was faster than for all TapGazer variants ($t(18) \geq 3.89, p \leq .002^{**}$). For TapGazer (Figure 6.7b), the main effect of Gaze ($F(1, 19) = 7.84, p = .012^{*}$) was significant with a 'small to medium' effect size ($\omega^2 = 0.04$), indicating that taps with gaze were faster. The main effect of Completion ($F(1, 19) = 0.01, p = .92$) and the interaction effect ($F(1, 19) = 0.28, p = .61$) were not significant.

Selection Time. Figure 6.8c shows the average the average times taken to select a candidate in Tapgazer after tapping a word's input string. The main effects of Gaze ($F(1, 19) = 1.19, p = .29$) and Completion ($F(1, 19) = 4.37, p = .05$) and the interaction effect ($F(1, 19) = 0.01, p = .93$) were not significant.

Search Time. Figure 6.8d shows the average times taken between tapping a word's input string and confirming the choice of a candidate with a tap, approximating visual search and think time by not counting the taps involved in selection. Gaze selection requires an additional right thumb tap, and manual selection requires an additional right thumb tap and a thumb/index/middle finger tap, which were adjusted for by subtracting the respective participant tap time averages. The main effects for Gaze ($F(1, 19) = 7.33, p = .014^{*}, \omega^2 = 0.11$) and Completion ($F(1, 19) = 7.99, p = .01^{**}, \omega^2 = 0.05$) were significant with 'medium' effect sizes, indicating that candidate search was

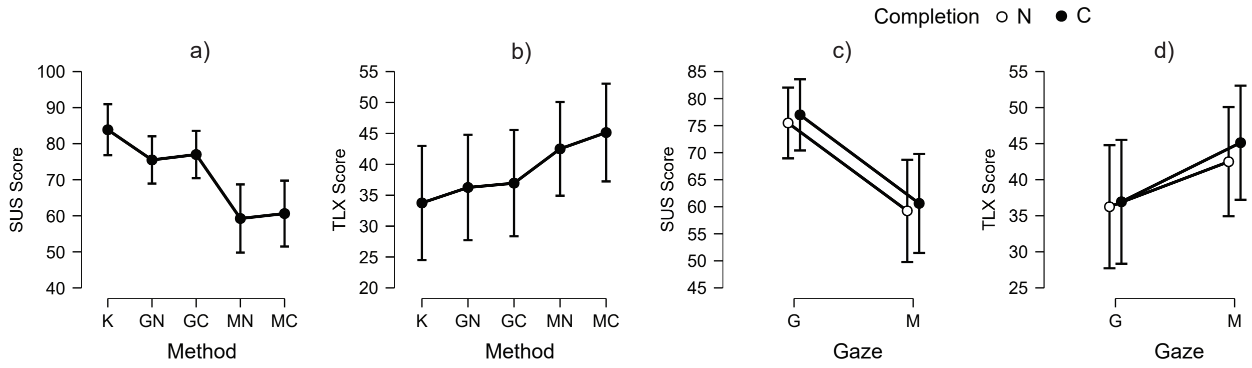


Figure 6.9: Evaluation results for SUS and TLX comparing QWERTY keyoard (K) with TapGazer in GN, GC, MN, and MC modes.

faster with gaze selection and without word completion, respectively.

Error Rate. Figure 6.7c shows the average total error rate for the five conditions. Users typed at $M = 8.47\%$ ($SD = 0.025$) for K and $M \leq 4.60\%$ for TapGazer. The main effect of Condition ($F(4, 76) = 29.85, p < .001^{***}$) was significant, and the main effect of Session ($F(4, 76) = 1.24, p = .30$) was not significant. All TapGazer variants had significantly lower error rates then K ($t(18) \geq 8.14, p < .001^{***}$). For TapGazer (Figure 6.7d), the main effect of Gaze ($F(1, 19) = 3.16, p = .09$) and Completion ($F(1, 19) = 0.20, p = .66$) and the interaction effect ($F(1, 19) = 0.23, p = .64$) were not significant.

Usability. Figure 6.9a shows the average SUS usability scores for the five conditions, with $M = 83.88$ ($SD = 15.12$) for K, about $M = 78$ ($SD = 14$) for GN and GC, and about $M = 61$ ($SD = 19.55$) for MN and MC. TapGazer with manual selection had significantly lower SUS scores then the other conditions ($t(18) \geq 3.75, p \leq .002^{**}$), and the differences between TapGazer with gaze selection and K are not significant ($t(18) \leq 2.1, p \geq .15$). For TapGazer (Figure 6.7c), the main effect of Gaze ($F(1, 19) = 24.00, p = .001^{***}$) was significant. The main effect of Completion ($F(1, 19) = 0.47, p < .50$) and the interaction effect ($F(1, 19) = 0.001, p = .97$) were not significant.

Workload. Figure 6.9b shows the average NASA-TLX task load scores for the five conditions, with $M = 33.75$ ($SD = 19.73$) for K, about $M = 36$ ($SD = 18$) for GN and GC, $M = 42.50$ ($SD = 16.18$) for MN, and $M = 45.14$ ($SD = 16.91$) for MC. MC had significantly higher task load scores than K ($t(18) \geq 3.18, p \leq .02^{*}$); the other differences are not significant ($t(18) \leq 2.48, p \geq .14$). For TapGazer (Figure 6.7d), the main effect of Gaze ($F(1,19) = 6.23, p = .02^{*}$) was significant with a 'medium' effect size ($\omega^2 = 0.04$), indicating that gaze selection has a lower task load. The main effect of Completion ($F(1,19) = 1.00, p < .33$) and the interaction effect ($F(1,19) = 0.28, p = .61$) were not significant. We analysed the six dimensions of task and in NASA-TLX separately and found that the only significant effect was the main effect of Gaze on Frustration ($F(1,19) = 11.98, p = .003^{**}$), indicating that gaze selection was less frustrating.

Preferences and Qualitative Feedback. When asked about which variant of Tapgazer they preferred, 15 of the 20 participants preferred GC the most, and 5 preferred GN the most. In the post-interviews, participants were overall positive about Tapgazer ("*I can type very fast after practice*", "*save energy by just tapping without reaching the specific letter*"). Several participants stated it was easy to find the right candidate words ("*the candidate words are different from each other and easy to locate the word to type*", "*look at where the word will show up and select it when it shows*"). Several participants noted that gaze selection was more usable than manual selection ("*it is really convenient when using gaze mode*", "*selecting words with gaze is more comfortable. It is easy to make mistakes while using manual selection mode*", "I need to think about which finger to tap when using manual selection"). Most participants appreciated the ability to complete words ("I'd love to have more words to select from", "*can save quite a few keystrokes when using TapGazer with completion*", "*I want to scan all candidates words in case I found the correct one*"), but some noted that it was easier not to consider completion of words ("*focus on typing the word and no need to worry about the candidates shown. And usually, I don't need to type the entire word when it is long*" – referring to the fact that in the TapGazer variants without word completion, the most likely incomplete candidate is shown in absence of complete candidates, allowing users to quickly complete long words). All participants reported that they would be willing to use TapGazer for off-desktop scenarios like VR.

6.7.1 Discussion

Our results demonstrate that overall, TapGazer is an easy-to-use system that can reach higher average WPM speeds than other text entry methods addressing similar use cases if gaze selection is used (see Table 6.1, on average 52.17 WPM as opposed to 44.6 for the best competitor). Furthermore, TapGazer achieves significantly lower error rates than the QWERTY keyboard, as word-level text entry avoids some sources of error of character-level text entry: while QWERTY typing, participants frequently used the right finger on the wrong key – a mistake that does not affect TapGazer. The higher WPM averages listed in Table 6.1 are mainly for experienced 'expert' users; however, our participants were all novice users of TapGazer. There were significant trends of improvement over the sessions, indicating that even better performance could be achieved with more practice. TapGazer with manual selection performed markedly worse; the extra tap required for selection slightly breaks the normal rhythm of QWERTY typing and forces participants to think about finding their word as well as using the right finger for selection. However, it is still very competitive compared to the related works in Table 6.1, and with practice using the right finger to select the appropriate candidate will likely become automatic for a user. Tapgazer with and without word completion performs similarly and both have their place; some users prefer to just type and not think about the completion of words, while others prefer to look for incomplete candidates before completing an input string. We analyze these two strategies further in subsection 6.7.2.

Limitations. The global COVID-19 crisis forced us to perform TapGazer's evaluation remotely. This necessitated three compromises: 1) Tapping was performed on a physical keyboard as opposed to a passive touch surface, providing active feedback through the mechanical keys; 2) although TapGazer was designed for HMD users, our participants did not have HMDs; and 3) most participants did not have access to a gaze tracker so that we had to simulate selection with a gaze. To address the first point, we performed a pilot study comparing TapGazer performance on a physical keyboard vs. on a passive touch surface (see subsection 6.7.2). The results indicate that the differences between the two are minor, so our results do likely generalize to tapping on other surfaces. In order to address the lack of HMDs, we went through the webcam footage recorded from all experimental sessions and confirmed that users did indeed keep looking at the screen while typing, as instructed, and not at their hands or elsewhere. This indicates that they were visually immersed in the TapGazer interface, in keeping with the experience one would expect in an HMD. In order to address the validity of simulating gaze for candidate selection, we analyzed candidate selection in more detail: First, the difference in selection time for gaze vs. manual was not significant, so it is unlikely that participants would have gained a lot by 'cheating' and not looking at the right candidate. Half of the TapGazer data we collected is based on manual selection, and the manual selection time results for word completion vs. non-completion are consistent with those of gaze selection (roughly the same difference); we would have expected word completion to be more affected by 'cheating' as it would have provided more opportunities. Finally, 'cheating' would have resulted in a higher error rate for gaze, especially in GC, as participants not looking for the right candidate would likely have selected before the right candidate was shown. The tap logs confirm that participants used word completion frequently in GC and that they would have made a lot more errors if they had simply tapped the right thumb without making sure the right candidate was present. The error rate is overall very low, and almost the same for GC and GN, which supports the assumption that our simulation of gaze was accurate. Another limitation is the fact that our study uses novices and is mainly cross-sectional, thereby likely underestimating the average performance of longer-term users. Finally, the study evaluates TapGazer only with participants seated at a desk, while our design aims for TapGazer to be usable in a variety of poses, e.g. standing. However, our setup is similar to the way most related works evaluated their performance (Table 6.1), so this makes our results more comparable to those of others.

6.7.2 Model-based Analysis

In the following, we will discuss models describing the user performance of TapGazer and its variants (RQ3) and then apply them to the analysis of design options and usage strategies. Because TapGazer is based on QWERTY typing, it is plausible to estimate performance based on a user's QWERTY typing speed. Based on the data from Section 6.7, wpm_K is significantly positively correlated with WPM_{GN} ($r(18) = 0.66, p < .001^{***}$), WPM_{GC} ($r(18) = 0.71, p < .001^{***}$), WPM_{MN} ($r(18) = 0.53, p = .008^{**}$), and WPM_{MC} ($r(18) = 0.58, p < .004^{**}$), with 'large' effect sizes [Cohen 1992]. Linear slope regression analysis yielded significant regression equations: $WPM_{GN} = 0.77 \times wpm_K$, $WPM_{GC} = 0.76 \times wpm_K$, $WPM_{MN} = 0.58 \times wpm_K$, and $WPM_{MC} = 0.54 \times wpm_K$. This confirms that there is a strong linear relationship between QWERTY typing and TapGazer performance, with users that are fairly new to TapGazer achieving on average 77% of their QWERTY typing speed when using the GC variant. Similarly, the average time taken for typing a key $type_K$ is significantly positively correlated with the average times for tapping a finger tap_{GN} ($r(18) = 0.80, p < .001^{***}$), tap_{GC} ($r(18) = 0.84, p < .001^{***}$), tap_{MN} ($r(18) = 0.76, p < .001^{***}$), and tap_{MC} ($r(18) = 0.80, p < .001^{***}$). Linear slope regression analysis yielded significant regression equations: $tap_{GN} = 1.31 \times type_K$, $tap_{GC} = 1.32 \times type_K$, $tap_{MN} = 1.52 \times type_K$, and $tap_{MC} = 1.48 \times type_K$.

Input Device. Our evaluation asked users to tap on a physical QWERTY keyboard, giving them flexibility as to which keys they tap by assigning keys to fingers according to their fingers' typical ranges of movement. However, TapGazer was designed to work without a physical keyboard, allowing users to tap on any surface. Therefore we need to analyze how far TapGazer performance changes if, instead of a physical keyboard, we use an input device that can pick up finger taps on a surface directly. Such a device could use sensors in the surface (e.g. adequately-sized touchpads or touchscreens), sensors on the fingers (e.g. TapStrap), sensors along the neural or muscular pathways controlling the fingers (e.g. wrist EMG), or optical tracking (e.g. Leap Motion).

To model the effect of using a flat-surface tap input device on TapGazer performance, we conducted a small study (n=6) comparing TapGazer typing skills (using gaze and word completion) on a physical QWERTY keyboard vs. touchpad surfaces (two Sensel Morphs). In both conditions, participants wore a Tobii HTC Vive Devkit VR headset to simulate a realistic use case for TapGazer (Fig. 6.3a). Participants were seated at a desk and first performed a 1-minute standard QWERTY typing test. Then participants put on the headset, calibrated the gaze tracker, and ran the official Tobii gaze demo to get used to the headset. They were then shown how TapGazer works and given time to practice it until they were able to correctly type at least 5 phrases taken from the MacKenzie corpus [MacKenzie and Soukoreff 2003]. During this training phase, we made sure that TapGazer was configured according to the finger-to-key mapping they preferred. Next, the two 5-minute typing sessions were performed, one for TapGazer on a physical keyboard and one for TapGazer with the Sensel Morphs. Participants were able to move their fingers freely on the Sensel Morph touch surface and we were able to identify each finger by its relative position. A short break was allowed between the conditions, the gaze tracker was recalibrated for the second condition, and the order of conditions was counterbalanced. Our participants (2 female, 4 male) varied in their QWERTY touch typing abilities (wpm range 28.6 - 72.5, average 50.8 wpm), two were wearing glasses, and half of them had never used eye-tracking devices before. We used correlation and regression analysis to build and validate a linear model of TapGazer speed for a touch surface WPM_{TS} based on TapGazer speed for a physical keyboard WPM_{TK}. wpm_TS is significantly positively correlated with WPM_{TK} ($r(4) = 0.86, p = .01^*$) with ‘large’ effect size. Linear slope regression analysis yielded a significant regression equation: $WPM_{TS} = 0.98 \times wpm_{TK}$. This indicates that the speed of tapping on a keyboard (as opposed to typing on a keyboard) is comparable to the speed of tapping on a passive surface, so it is likely that our evaluation results can be generalized to other suitable input devices.

Slow Typists. Some people are slow typists, e.g. when they are just learning to type. Word completion can be particularly useful for them. This is similar to text entry on a mobile phone, where tapping individual keys can be slow and many people use word completion extensively to speed up text input. In the following, we show how to estimate the QWERTY typing speed that marks the point in typing and tapping skill where not using word completion becomes faster than using word completion with a visual search for the correct word after every tap.

Similar to the well-known Keystroke-Level Model (KLM) [Card et al. 1980], we model the time T_{GN} required for entering a word w in TapGazer without word completion based on: a) the average tapping time for fingers tap; b) the average tapping time for the right thumb $space$ (which types space in the standard QWERTY mapping), and c) the average visual search time $search_{GN}$ for finding the desired word among the completed words shown: $T_{GN}(w) = |w| \times tap + search_{GN} + space$. That is, we sum up the average tapping time for each of the $|w|$ letters, the average search time for spotting the right completed word, and the average time of the confirmatory tap with the thumb. Note that by definition, this model predicts the average word completion time for our evaluation of GN exactly when substituting our measured average values for the model parameters. Similarly, we model the time T_{GC} required for entering a word w in TapGazer with word completion, assuming that the user looks at the suggested words after every tap. This time we consider the number of taps $c(w) \leq |w|$ required until w appears in the list of suggestions, and the average visual search time $search_{GC}$ for finding a desired word among suggested, possibly incomplete words: $T_{GC}(w) = c(w) \times (tap + search_{GC}) + space$. The model illustrates the trade-off between a reduced number of taps and increased time spent per tap.

In Section 6.7.2 we have shown that there is a strong linear relationship between tapping and QWERTY typing speed. Therefore, in order to estimate T_{GN} based on the time T_K required to type word w, we substitute tap and $space$ by corresponding linear estimates $1.31 \times type_K$ and $0.91 \times type_K$, respectively. Because search times do not vary with QWERTY speed, we approximate them using averages $search_{GN} = 148$ ms and $search_{GC} = 420$ ms. The latter is the average for GC when the maximum number of candidates is shown (10) so that it is not immediately apparent which candidate to choose, as this is the most likely case when looking for word completions after every tap. Furthermore, we substitute the average word length in English text $|w| = 4.79$ [Norvig 2013], and the expected number of taps $c = 2.41$ required before the desired word appears in the suggestions. The latter was determined using a simulation of the word-frequency based suggestion algorithm used in GC on a dictionary of the 7,582 most frequent English words. This results in estimates of the average times per word dependant on $type_K$: $T_{GN} = 7.18 \times type_K + 148$ ms and $T_{GC} = 4.07 \times type_K + 1012$ ms. T_{GN} and T_{GC} are equal at $type_K = 278$ ms, which is equivalent to about $wpm_K = 37$. Therefore, typists much slower than that would likely be faster using TapGazer with word completion. A better word prediction algorithm will reduce the expected value for $c(w)$, increasing the estimated speed at which word completion becomes a hindrance. A similar analysis can be made for the non-gaze alternatives of TapGazer MN and MC.

Power Users. If the prediction algorithm used to generate suggestions for word completion is reasonably stable, i.e. if users can anticipate which word will be suggested as the most likely option, then power users will learn for frequent words how many taps they need before they can simply accept the most likely suggested word. In both GC and MC, the most likely suggestion can be quickly accepted without even looking at the word suggestions, by tapping the right thumb. Let us assume a power user has learned all the prefixes that must be tapped to make each of the 100 most frequent words of the English language the most likely suggestion, e.g. “tapping ‘t’ makes ‘the’ the most likely word.” According to our word frequency data, the 100 most common words account for 48.12% of all English texts. Let $c(w)$ be the number of taps a user needs to do before the word suggested as most likely is the desired word w. Similar to Section 6.7.2, this leads to the following model for a power user who uses word completion without visual search for the 100 most frequent words (first summand) and types words in full otherwise (the second summand, using $search_{GN}$ as the search is only among the completed words, which come first):

$T_{GC}(w) = 48.12\%(c(w) \times tap + space) + 51.88\%(|w| \times tap + search_{GN} + space)$.

According to our simulation of the word-frequency based suggestion algorithm used in GC and MC, which is based on the 7,582 most frequent English words, the expected number of taps a user needs to make before one of the 100 most frequent words becomes the most likely suggestion is $c = 2.05$. This is lower than one might think, as the three most frequent words (the, of, and), which account for more than 14% of English texts, all use different fingers on their first tap, so each appears immediately as a most likely suggestion. Furthermore, our simulation reveals that six of the 100 most frequent words (my, or, if, now, our, then, go) are never shown as the most likely suggestion; they typically make up 1.15% of English texts, therefore we shift this percentage from the first to the second summand in our model. As in Section 6.7.2, we substitute c, the average word length in English texts $|w| = 4.79$, and estimates of $search_{GN}$, tap and $space$. To relate the model to QWERTY typing speed, we describe tap and $space$ as linear estimates of $type_K$. The result is $T_{GC} = 5.5 \times type_K + 78$; the corresponding W_{GC} can be approximated for typical QWERTY typing speeds (up to 80 wpm) with a linear lower bound of $WPM_{GC} = 0.95 \times wpm_K$ (compared to $0.76 \times wpm_K$ for novice users). By learning tap prefixes for frequent words so that these words can be selected quickly without visual search, TapGazer is expected to allow power users to achieve typing speed close to QWERTY typing. Even if a user learns tap prefixes only for the 10 most common words, this accounts for about 25.13% of English texts and the estimated speed is $WPM_{GC} = 0.82 \times wpm_K$. Power users can use the same approach for TapGazer without gaze tracking (MC), with estimates $WPM_{MC} = 0.78 \times wpm_K$ when learning prefixes for the 100 most frequent words and $WPM_{MC} = 0.65 \times wpm_K$ for the 10 most frequent words (compared to $0.54 \times wpm_K$ for novice users). When using gaze tracking, if a power user furthermore learns where a frequent word appears for the first time in the suggestions, e.g. "after tapping the left ring finger 'with' appears at the center-left", then the power user could potentially look at the right suggestion and select it immediately, reducing $c = 1.28$ and leading to an estimate of $WPM_{GC} = 1.03 \times wpm_K$ for the 100 and $WPM_{GC} = 0.84 \times wpm_K$ for the 10 most frequent words. If a power user is willing to learn a new layout, i.e. a finger-to-letter mapping not based on QWERTY, then c can be reduced further. We used branch-and-bound search to find a mapping that minimizes c for the 100 frequent words, resulting in mapping with $c = 1.18$ and $WPM_{GC} = 1.05 \times wpm_K$ for learned prefixes only, if the positions of the respective word suggestions are also learned. In summary, learning tap prefixes and even display positions for common words can potentially speed TapGazer up drastically, with and without gaze tracking.

6.7.3 Discussion

Similar to KLM [Card et al. 1980], our models are based on the average measurements obtained from the evaluation. As a result, their predictions will be inaccurate to some degree when applied to different groups of users. In particular, our experiments collected TapGazer performance data only from novice users. A multi-level regression analysis shows that the effect of session number on wpm was significant ($B = 1.99, 95\%CI = [1.50, 2.47], t(398) = 8.09, p < .001^{***}$), indicating that participants increased their wpm by about 2 wpm for each usage session. Users will likely continue to get faster with practice. The models we created based on short term use are therefore likely to underestimate the performance of longer-term users, forming a reference baseline for future research. Also, the models add value by formalizing strategies that some users will likely apply to increase their TapGazer performance. Finally, the models identify important parameters affecting TapGazer's performance, providing starting points for further improvements in future work.

6.8 Conclusion

We have presented TapGazer, a novel text entry method combining tapping and gaze. TapGazer was designed to facilitate text entry while wearing VR/AR HMDs, without the need for a physical keyboard or to look at one's hands, in anticipation of a future where affordable AR glasses will be as ubiquitous as mobile phones are today. Our results indicate that novice users can achieve 77% of their QWERTY typing speed, with an average TapGazer WPM of 52.17, which surpasses the performance of comparable text entry methods. Furthermore, the error rate of TapGazer is significantly lower than for a physical QWERTY keyboard. We have created performance models for TapGazer that illustrate how different users can benefit from different usage strategies, and identify important performance parameters that can be optimized in future design iterations. In future work, we anticipate longer-term studies of performance, the use of Tapgazer with different devices and in different poses, improvements to word prediction based on more sophisticated language models, and error correction.

Design	WPM	Examples
Tapping QWERTY on a touch surface	17.2–44.6	BlindType [Lu et al. 2017], TOAST [Shi et al. 2018], PalmBoard [Yi et al. 2020]
Gesture typing	16.0–42.7	GestureType [Yu et al. 2017], Chen et al. [Chen et al. 2019], KeyScretch [Costagliola et al. 2011]
Typing on tiny surface	26.0–41.0	Vertanen et al. [Vertanen et al. 2018], VelociTap [Vertanen et al. 2015], Ahn & Lee [Ahn and Lee 2019]
Mid-air chord gesture typing	22.0	Adhikary [Adhikary 2018], Sridhar et al. [Sridhar et al. 2015]
Typing with pinch gestures	11.9–23.4	BiTipText [Xu et al. 2020], DigiTouch [Whitmire et al. 2017], TipText [Xu et al. 2020]
Mid-air finger tapping	17.8–23.0	ATK [Yi et al. 2015], VISAR [Dudley et al. 2018]
Reduced physical QWERTY keyboard	7.3–30.0	Stick [Green et al. 2004], 1Line [Li et al. 2011], LetterWise [MacKenzie et al. 2001], VType [Evans et al. 1999]
Tapping with head or controller on a soft alternative keyboard	10.2–21.1	RingText [Xu et al. 2019b], PizzaText [Yu et al. 2018], HiPad [Jiang and Weng 2020], Boletsis & Kongsvik [Boletsis and Kongsvik 2019], Curved QWERTY [Yanagihara et al. 2019]
Tapping QWERTY with head or controller	11.3–15.6	Tap/Dwell [Yu et al. 2017]
Gaze typing plus touch	14.6–15.5	EyeSwipe [Kurauchi et al. 2016; Kurauchi 2018], TAGSwipe [Kumar et al. 2020]

Table 6.1: Summary of prior text entry solutions for English words that are compatible with our usage scenarios.

7 Web-based Video Conferencing System with Gaze-awareness and Spatial information

7.1 Introduction

Lastly, I extend the idea from immersive spaces to real-life environment, especially because of such a different working and living style in year 2020. I want to investigate how to convert my expertise to contribute to experience with minimal setup.

Video conferencing is becoming the dominant medium for remote discussion and collaboration during the global COVID-19 pandemic. However, *gaze awareness information for more than two people* in mainstream video chat tools such as Skype, Zoom, and Google Meet is insufficiently conveyed through this medium. Hence, it is almost impossible to use eye gaze as a nonverbal cue to infer where attention is directed in conventional video conferences.

Prior art in remote small-group collaboration has leveraged multi-view cameras, customized displays, or mixed-reality settings to solve this problem. For example, GAZE-2 [Vertegaal 1999; Vertegaal et al. 2003] employs an array of cameras with an eye tracker and selectively transmits the preferred video stream to remote users. MMSpace [Otsuka 2016] introduces novel physical kinetic displays to support gaze awareness between every pair of participants; Holoportation

[Orts-Escolano et al. 2016] leverages full-body reconstruction and headset-removal technologies to achieve immersive small-group telepresence with Microsoft HoloLens. However, it is still unclear how to embed eye contact in a conventional videoconference with an ordinary laptop; or to determine the potential benefits and drawbacks of visualizing gaze awareness in remote small-group conversation.

In this chapter, we present and evaluate LookAtChat, a video conferencing system that visualizes gaze awareness for remote small-group conversation. LookAtChat consists of three components: a WebRTC server to support videoconferencing and logging, an eye-tracking module powered by `WebGazer.js` [Papoutsaki et al. 2016] to recognize gaze positions, and a visualization module implemented with the `three.js`[1].

As initial work, the research questions are exploratory: How do people perceive eye contact in conventional video conferences? Can visualization of eye contact improve remote communication efficiency? In what context will users prefer to see eye contact? What forms of eye-contact visualization may be preferred by users?

To create LookAtChat, we conducted formative interviews with five people who use videoconferencing with colleagues daily. Our research is inspired and informed by prior small-group communication systems that demonstrate the potential of visualizing gaze awareness and spatial information. To improve the generalizability and replicability of the system, we only require each user to use a laptop with a webcam. We further extend the design space of visualizing gaze and spatial information to a total of 11 layouts.

To evaluate LookAtChat, we conducted four three-session user studies with 20 remote participants (ages 24-39, 6 female and 14 male). In our analyses of video recordings, post-activity questionnaires, and post-hoc interviews, we found that LookAtChat can effectively engage participants in small-group conversations by visualizing eye contact and providing spatial relationships. Gaze and spatial information can improve the conversation experience, bring greater social

[1] three.js: JavaScript 3D library, http://www.threejs.org.

presence and richness, and provide better user engagement.

7.2 Related Work

To understand how gaze information is integrated into video conferences and to justify our design decisions, we review prior art on multi-user experience in distributed collaboration and gaze tracking technologies for videoconferencing. Many researchers have contributed to investigating future workspaces such as improving individual productivity like HoloDoc [Li et al. 2019], reconstructing multi-user experiences like "the office of the future" [Raskar et al. 1998], and cross-device interaction [Voelker et al. 2020]. Furthermore, remote conferencing shows its potential for geologically dispersed users and is efficient for group discussion [Neustaedter et al. 2015]. In scenarios requiring certain levels of trust and judgment with non-verbal communication, non-verbal cues are highly important for effective communication [Regenbrecht and Langlotz 2015]. Gaze support and feeling of face-to-face [Olson et al. 1995] play a central role in those scenarios. With the increasing development of gaze tracking devices and technology, gaze-assisted interaction is becoming popular in the fields of text entry [Ward et al. 2000], video captions [Kurzhals et al. 2020], and video conferences.

7.2.1 Multi-user Collaboration in Distributed Environments

Distributed multi-user collaboration has been widely researched from the perspective of locomotion, shared proxies, and life-size reconstruction as well as different purposes including communication, presentation, and object manipulation. Your Place and Mine [Sra et al. 2018b] creates experiences that allow everyone to walk in collaborative VR. Three's Company [Tang et al. 2010] presents a three-way distributed collaboration system that places remote users either on the same side or around a round table. Besides, Three's Company provides non-verbal cues like body gestures through a shared tabletop interface. Remote users' arm shadows are displayed locally on a tabletop device, which is beneficial for collaborative tasks with shared objects. Tan *et al.* [Tan et al. 2010] focus on presentation in large-venue scenarios, creating a live video view that seamlessly combines the presenter and the presented material, capturing all graphical, verbal, and nonverbal channels of communication. Tele-Board [Gumienny et al. 2011] enables regionally separated team members to simultaneously manipulate artifacts while seeing each other's gestures and facial expressions. The concept of Blended Interaction Spaces [O'hara et al. 2011] is proposed to providing the illusion of a single unified space by creating appropriate shared spatial geometries. TwinSpace [Reilly et al. 2010] is a generic framework discussing brainstorming and presentation in cross-reality that combines interactive workspaces and collaborative virtual worlds with large wall screens and projected tabletops. Physical Telepresence workspaces [Leithinger et al. 2014] is a shaped display providing shape transmission that can manipulate remote physical objects. Cameras are widely used for the above alternatives to capture users, besides, 360 videos have recently been researched. SharedSphere [Lee et al. 2018] is a wearable MR remote collaboration system that enriches a live captured immersive panorama based collaboration through MR visualization of non-verbal communication cues.

Immersive collaborative virtual environment (ICVE) and Augmented Reality (AR) can be used to develop new forms of teleconferencing, which often leverages multiple cameras set up and 3D reconstruction algorithms. EyeCVE [Steptoe et al. 2008] uses mobile eye-trackers to drive the gaze of each participant's virtual avatar, thus supporting remote mutual eye-contact and awareness of others' gaze in a perceptually coherent shared virtual workspace. Jones *et al.* [Jones et al. 2009] design a one-to-many 3D teleconferencing system able to reproduce the effects of gaze, attention, and eye contact. A camera with projected structure-light is set up for reconstructing the remote user. Billinghurst and Kato [Billinghurst and Kato 2000] developed a system that allows virtual avatars and live video of remote collaborators to be superimposed over any real location. Remote participants were mapped to different fiducial markers. The corresponding video images were attached to the marker surface when markers are visible. Room2Room [Pejsa et al. 2016] is a telepresence system that leverages projected AR to enable life-size, face-to-face, co-present interaction between two remote participants by performing 3D capture of the local user with RGBD cameras. Holoportation [Orts-Escolano et al. 2016] demonstrates real-time 3D reconstructions of an entire space, including people, furniture, and objects, using a set of depth cameras. Gestures are preserved via full-body reconstruction and headset removal algorithms are designed to convey eye contact. However, "uncanny valley" remains a challenging problem in this domain.

7.2.2 Eye Contacts and Gaze Correction Technology in Video-mediated Conversation

Various hardware setups have been explored for gaze correction including hole in screen, long-distance, and half-silver mirror. The hole in screen concept is about drilling a hole in the screen and placing a camera. Long-distance uses a screen at a far distance while placing the camera as close as possible [Tam et al. 2007]. The half-silver mirror allows a user to see through a half-transparent mirror while being observed by a well-positioned camera at the same time. This idea was adapted in ClearBoard [Harrison et al. 1995; Ishii and Kobayashi 1992] and Li *et al.*'s transparent display [Li et al. 2014]. Despite their advantages in terms of system complexity and costs, such solutions are rarely used outside of the laboratory due to the availability of hardware. In the meantime, quite a few 2D video-based (or image-based) approaches are proposed for eye contact including eye correction with a single camera [Andersson et al. 1996; Andersson and Chen 1997] and multiple cameras [Criminisi et al. 2003] while applying image-based approaches like texture remapping and image warp [Gemmell et al. 2000]. However, the technology is not sufficiently accurate to avoid visual artifacts and the uncanny valley. 3D video-based solutions including 3D reconstruction is another trend for maintaining eye contact while the head is reconstructed. RGB camera [Xu et al. 1999], depth camera [Zhu et al. 2011], Kinect [Kuster et al. 2012], or motion capture system [Maimone et al. 2013] are used for 3D reconstruction.

Eng *et al.* [Eng et al. 2013] propose a gaze correction solution for a 3D teleconferencing system with a single color/depth camera. A virtual view is generated in the virtual camera location with hole filling algorithms. Compared to a single-camera setup, multiple cameras are popularly used for providing gaze [Ashdown et al. 2005] in videoconferencing. True-view [Xu et al. 1999] was implemented with two cameras (one on the left and the other on the right). The synthesized virtual camera view image at the middle viewpoint is generated to provide correct views of each other and the illusion of proximity. GAZE-2 [Vertegaal 1999; Vertegaal et al. 2003] utilizes an eye tracker with three cameras. The eye tracker is used for selecting a proper camera closest to where the user is looking. GAZE-2 prototypes an attentive virtual meeting room to experiment with camera selection. In each meeting room, each user's video image is automatically rotated in 3D toward the participant he is looking at. All the video images are placed horizontally so the video image turns left or right when the corresponding camera is chosen. Likewise, MultiView [Nguyen and Canny 2005, 2007] is a video conferencing system that supports collaboration between remote groups of people with three cameras. Additionally, MultiView allows multiple users to be co-located in one site by generating a personal view for each user even though they look upon the same projection surface, which they achieve by using a retro-reflective material. Photoportals [Beck et al. 2013; Kunert et al. 2014] groups local users and remote users together through a large display. All users are tracked and roughly reconstructed through multiple cameras and then rendered within a virtual environment. MMSpace [Otsuka et al. 2012, 2013; Otsuka 2016, 2017] provided realistic social telepresence in symmetric small group-to-group conversations through "kinetic display avatars". Kinetic display avatars can change pose and position by automatically mirroring the remote user's head motions. One camera is associated with one transparent display. Both camera and display can be turned to provide corresponding video input image and output angle. Sirkin *et al.* [Sirkin et al. 2011] developed a kinetic video conferencing proxy with a swiveling display screen to indicate which direction in which the satellite participant was looking for maintaining gaze and gestures to mediate the interaction. Instead of rendering a video image on a rectangular display, a cylinder display is proposed in TeleHuman [Kim et al. 2012] with 6 Kinects and a 3D projector.

LookAtChat is designed to be used with *a minimum requirement of a laptop/PC and a single webcam*. While multi-view cameras and external hardware may yield better eye tracking and 3D rendering solutions, such systems typically require very high computational power and exclusive hardware setups. Since it is possible for users with low-cost video conferencing set up to learn to interpret gaze direction to a very high degree of accuracy [Grayson and Monk 2003], we decided not to apply extensive image-based manipulation on video streams but rather to focus on the design of a widely accessible online system to empower video conferencing users with real-time visualization of gaze awareness.

7.3 Design

To inform the design of LookAtChat and understand whether and how gaze information affects video conferencing, we conducted five formative interviews with video conferencing users (2 female and 3 male, labeled as I1 to I5) to learn the advantages and disadvantages of current videoconferencing software compared to real-life meetings as well as people's expectation of videoconferencing. We asked participants about their recent video conferencing experience under different scenarios. Our takeaways are summarized below.

7.3.1 Formative Interviews

Good for multi-tasking and information sharing.

Software such as *Zoom* and *Skype* allows participants to work on multiple tasks at the same time while video conferencing, such as walking on a treadmill while listening to a talk. Users benefit from sharing screen or notes through video conferencing software, as it allows any participant to instantly share their own document or presentation. Although participants in offline meetings can share information through whiteboarding or printed documents, video conferencing software allows a large number of people to concentrate on the same document and work on different sections of it.

Bad for white-boarding and body gestures.

For group discussions in which all participants may need to contribute their thoughts, a physical whiteboard is very popular. And yet, shared free sketch software is not well integrated into video conferencing software or available as stand-alone software for now, though quite a few researches have focused on that in immersive environments. Similarly, body gestures are partially missing due to the small view area of cameras and missing/different spatial information of participants.

Bad for finding the speaking up timing.

P1 and P4 thought it was more difficult to know when to speak in online meetings because not all the participants' gaze and body information are perceived well through the camera. It is not clear who is talking to whom, whether the speaker is waiting for an answer from a specific person, or if a speaker is pausing or is ending the conversation during group discussion.

Bad for controlling meeting length.

The length of physical meetings is usually well controlled since the meeting rooms are usually booked throughout the day and participants are aware of those who are gazing through the window, waiting to use the room next. However, participants in virtual conferences often cannot find the best time to exit for the next meeting. I2(M) commented that "in virtual video conferences, very few people strictly follow the proposed length of the meeting and oftentimes delay the next meetings. People just keep talking when the meeting goes beyond the scheduled time".

Expectations of Future Videoconferencing Systems. **Improve the control of the conversation.** It is difficult to use words such as "you" in video conferencing contexts because participants barely know who the speaker is talking to, while “you” is natural in co-located conversations. I1(M) felt “less involved” because of the lack of this information. There also exists more simultaneous speech in video conferences. People start talking together and stop together to wait, which causes participants to lose track of the conversation.

Provide spatial information. I3(F) wished to select a seat the way they would normally enter a meeting room: “Everyone has their own perspectives and maintain the spatial relationship with each other”.

Thoughts of Visualizing Gaze Information in Videoconferencing System. **Good for natural discussion.** Interviewee (I2(M) and I5(F)) think it is helpful especially if the discussion requires feedback, attention, and interaction. Also, it helps branch ideas. It is easy to suggest what topic one participant is following by looking at the proposer directly in offline meetings, but not easy to show to the group information in videoconferencing software.

Different for small group and large scale. For presentations or lectures, presenters or teachers may benefit from participants' gaze information that helps them adjust content in real-time. P1 elaborated: “teachers know the topic is difficult or get distracted when quite a few students' gaze focuses are shifted.”

Concerns for privacy. Some interviewees (I3(F) and I5(F)) mentioned that they feel pressured when being looked at or looking at others. Displaying anonymous gaze information or aggregated data and reporting the result afterward may be helpful.

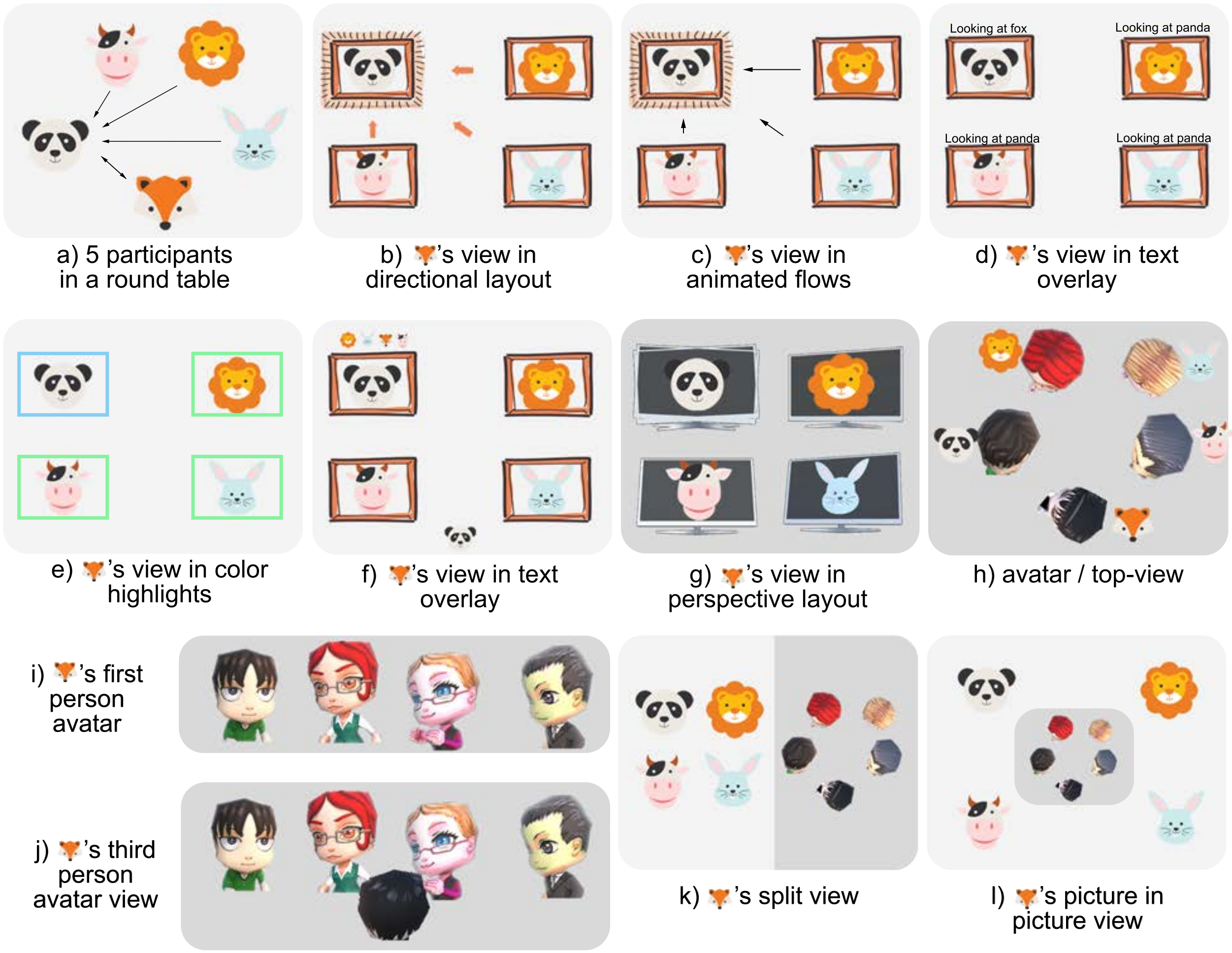
Informed by formative interviews and inspired by prior systems, we formulate our design rationale, explore the design space, elaborate on two specific layouts for natural integration with conventional videoconferencing, and discuss potential use cases.

7.3.2 Design Rationale

We constrain our design scope to remote small-group conversations in which all participants are physically dispersed. This setting is motivated by the circumstances of COVID-19, where everyone is working remotely. Users mostly participate in these conversations on a laptop with a built-in frontal camera or a workstation with a USB webcam. Scenarios with two or more people co-located in front of the camera for video conferencing are out of our design scope. Depth camera[Pejsa et al. 2016], multiple cameras[Orts-Escolano et al. 2016], motion capture systems[Maimone et al. 2013], professional eye-tracking systems[Vertegaal et al. 2003], or head-mounted displays[Orts-Escolano et al. 2016] are not considered as alternatives in our design due to their constraints of cost and availability. Although the above devices allow richer social engagements and more accurate gaze detection over a single webcam, we desire to make our platform accessible to most users. Taking these factors into account, we constructed a web application to prototype LookAtChat so that any device with a camera can access our website via a modern Internet browser such as Google Chrome.

7.3.3 Design Space

Elicited from formative interviews, we prototype LookAtChat to explore the design space visualizing gaze in videoconferencing. Considering popular video conferencing software is using 2D flat layouts for video image placement, we explore the design space on top of the traditional 2D flat layouts to expand the potential of 3D as well as hybrid layout alternatives. Hybrid dimension alternatives are proposed to combine 2D and 3D representations for taking advantage of both categories. The short description of our designs are listed in Table 7.1 with corresponding illustrations in Figure 7.1. Each sub-figure in Figure 7.1 illustrates a five-user scenario: user "panda" is speaking and looking at user "fox" while all the rest participants are listening to "panda" and looking at "panda".



a) 5 participants in a round table

b) 🦊's view in directional layout

c) 🦊's view in animated flows

d) 🦊's view in text overlay

e) 🦊's view in color highlights

f) 🦊's view in text overlay

g) 🦊's view in perspective layout

h) avatar / top-view

i) 🦊's first person avatar

j) 🦊's third person avatar view

k) 🦊's split view

l) 🦊's picture in picture view

Figure 7.1: Design space of LookAtChat. To convey eye contacts in a), 2D flat layouts (b – f), 3D immersed layouts (g - j), and hybrid layouts (k – l) illustrate the mutual gaze between “panda” and “fox” and how other participants gaze at “panda”.

7.3.4 2D Flat Layout

We first explore how to impose eye contact on a 2D flat layout. The visualization of eye contact on 2D flat layouts could be illustrated directly or indirectly. The direct design delivers straightforward signals with less cognitive load, while the indirect design may result in less interference with the video streams. Figure 7.1(b) and (c) are illustrated how *directional layout* and *animated flows* convey eye contact. From the perspective of user ‘fox’, ‘panda’ is looking at itself so the video frame is highlighted with outer glows. In the meantime, all other participants are looking at ‘panda’ so a static directional arrow is shown in Figure 7.1(b) and dynamic flow from observer to observee is rendered in Figure 7.1(c). *directional layout* applies fade-in and fade-out to indicate the start and end of gaze awareness in a smooth transition.

Figure 7.1(d), (e), and (f) demonstrate how *text overlay*, *color highlights* and *icon overlay* visualize eye contact indirectly. Figure 7.1(d) shows text overlays at the bottom of the video window. The names of observers are displayed following the FIFO rule. The first name in the list is the one who looks at the observee earliest. Figure 7.1(e) renders different color borders when different participants are looking at others. Likewise, Figure 7.1(f) illustrates the profile at the bottom of the video window. The profile is a thumbnail image of the corresponding user. We take the first frame of the video as a thumbnail reference.

2D flat layouts are widely adopted in commercial video conferencing software. We provided two levels of eye contact visualization: direct and indirect. Direct eye contact options demonstrate eye contact to users intuitively, so it helps users immediately understand gaze information on a subconscious level. Indirect eye contact options imply the eye contact subtly so users need to interpret the UI elements while the visual effects of elements are minimized so as not to be distracting.

7.3.5 3D Immersed Layout

We further investigate providing eye contact on 3D immersed layouts. The 3D immersed layouts are proposed to introduce spatial cues between participants as well as gaze information. The video image of all participants is attached to a monitor frame per user in the view. Instead of placing the video image directly in the scene and applying the perspective transformation, we employ the “physical kinetic displays”[Otsuka 2016] metaphor and attach the video image to a monitor frame. This helps users to perceive the video as a 3D display. Other alternatives such as painting frames or mirror frames can also replace the monitor frames. We also designed the 3D immersed layout with two different perspectives: first-person view and third-person view. Figure 7.1(g) and

(i) are first-person perspective designs. In *perspective layout*, users see a billboard representation of the video stream shaking if being looked at or turning if looking at other participants. Thus, we see user “panda” is shaking slightly in “fox”s view, and other users are turning to look at “panda” in Figure 7.1(g). In *avatar/first-person*, users see other participants as avatars representing their heads. The avatar will turn to the corresponding user when the user behind the webcam is looking at that user. So avatar ‘panda’ is facing the viewer (‘fox’) and other avatars are turning to look at ‘panda’ in Figure 7.1(i).

Figure 7.1(h) and (j) show the third-person perspective designs in the 3D immersed layouts. In *avatar / top-view*, all participants’ avatars are rendered from above and a real-time video texture is shown alongside. Users can infer the spatial relationships from the orientation of the avatars. In *avatar / third-person*, users’ cameras are placed behind their avatar so that each user can see other participants’ head orientation as well as their gaze cues.

The 3D immersed layout is designed to emphasize the spatial cues between participants. The effect is similar to 3D collaborative gaming experiences. We provide two levels of perspective and introduce 3D personalized avatars for user representation. Avatar representation is personalized according to real-time video textures (detailed in the next section). The spatial cues and gaze information is designed to be natural and similar to real-life scenarios, though it may introduce some sense of the uncanny valley when warping the video image to fit the 3D avatars.

7.3.6 hybrid Layout

We next explore the potential of hybrid layout designs that combine the 2D flat layout and 3D immersed layout. The hybrid layout is investigated to show a large video image and provide 3D gaze cues as well. We consider two rendering approaches: *split-view* (Figure 7.1(k) and *picture in picture* (Figure 7.1(l)). *split-view* organizes the 2D layout and 3D layout side-by-side. Users can perceive eye contact from the 3D layout while simultaneously viewing the other participants in the 2D layout. *picture in picture* allows users to acknowledge spatial cues at the center of their entire view. *split-view* allows users to choose the focus on either video texture or eye contact and spatial information. *picture in picture* prefers to present both pieces of information as a whole to users.

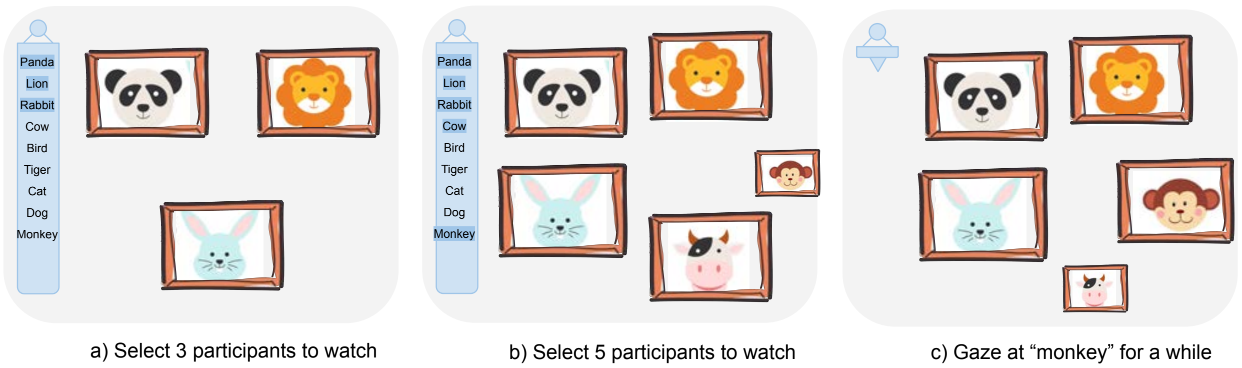
7.3.7 Directional Layout and Perspective Layout

As the first step towards visualizing gaze awareness for remote small-group conversations, we chose to implement and experiment with three conditions: baseline layout, 2D *directional layout* and 3D *perspective layout*. We have several considerations for selecting *directional layout* and *perspective layout* for comparison. Our overarching goal is to explore how gaze and spatial information facilitate video conferences. *directional layout* and *perspective layout* both show eye contact directly so that it is easy and straightforward for users to see and understand the system without further cognitive load. Also, we chose not to include avatar designs in the first experiment because it would have required higher graphics processing capabilities of users' computers than designs not including avatars. Also, the personalized avatars could likely introduce "uncanny valley" effects, which might negatively impact users' conversation quality. Lastly, hybrid layout designs are not selected for our exploratory user study, since we primarily want to understand how 2D flat layout and 3D immersed layout individually work for users.

7.3.8 Use Cases

Figure 7.1 demonstrates all the designs in a small-group discussion scenario. Meanwhile, video conferences are widely utilized in a large variety of use cases. LookAtChat is designed to be easily adapted to different video conferencing requirements.

Small-group discussion. Small-group discussion is one of the majority use cases in video conferencing, either for working or for entertainment purposes such as brainstorming, playing games, etc. To save network bandwidth and decrease the cognitive load of users, LookAtChat



a) Select 3 participants to watch

b) Select 5 participants to watch

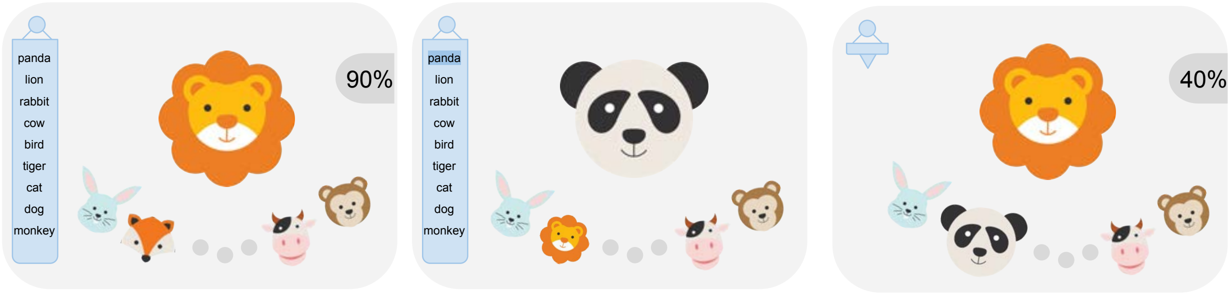
c) Gaze at “monkey” for a while

Figure 7.2: Video conferencing with varied numbers of participants in LookAtChat in 2D flat layout category. Size and placement of video image is updated in real-time according to the observer’s gaze.

provides a docked sidebar to show a full list of all participants. Users are free to choose a few participants of interest. The selected participants are rendered in a medium-size video image at the beginning. The size will grow or shrink depending on the user’s focus. Users can select or de-select participants at any time during the video conference (shown in Figure 7.2(a) and (b)). The participants who received focus from the user are gradually moved to the center Figure 7.2(b)(c).

Presentation. Slides presentation is another strong use case in video conferences. During presentations, presenters focused mostly on the slides or the shared windows and feedback from other participants. From formative interviews, people report being more interested in watching the slides than in watching the presenter. So LookAtChat may visualize participants’ focus with heatmaps. Presenters will see other participants’ gaze positions on the shared screen and listeners will see the presenter’s focus instead.

Large-scale meeting. Videoconferencing with a large audience is popular and useful for remote seminars and all-hands meetings. By default, the lecturer is rendered to all the participants (Figure 7.3(a)). Participants are free to see another participant as shown in Figure 7.3(b). We proposed to use aggregated data in this form. For example, Figure 7.3(a) shows how lecturers



a) lecturer's view: percentages of students who are paying attention to the lecture are listed by the lecture

b) a student (*fox*)'s view when panda volunteers to answer questions and obtains the most attention from the audience

c) lecture's view: users who receive more attention from the audience (e.g., panda) are highlighted with larger video frames.

Figure 7.3: Presentation with a large audience in LookAtChat. An aggregated number of gaze-received is shown to the lecturer (see the top-right percentages in a) and c)). b) Audience members can choose to watch others in a large view. Audience members who receive more eye contact have larger size video frames than others for the lecturer to pay attention to.

perceived other participants' looking at themselves and Figure 7.3(c) indicates when participants other than the lecturer receive focus.

7.3.9 Implementation

LookAtChat is designed and implemented for both video conferencing users and researchers to conduct remote user studies. LookAtChat comprises three major parts: a WebRTC server to support videoconferencing and hosting remote user studies, a real-time eye-contact detection module, and a WebGL-based renderer to visualize the gaze information.

Workflow. As Figure 7.4 demonstrates, LookAtChat employs a WebRTC server as well as peer-to-peer networking. For each newly-joined client, it talks to the WebRTC server (including Internet Connectivity Establishment server and Signaling) first to establish a peer-to-peer connection with existing clients. Hence, the clients can send and receive video and audio streams with each other. Next, LookAtChat server maintains the identifier of each client after the WebRTC connection is established. For each client, gaze and audio level information are processed locally and sent to LookAtChat server. Afterward, the server broadcasts the information to all of the clients and the renderer on the client-side will locally visualize the gaze and audio information.

Host Mode to Support Remote User Study Due to the challenges of the global COVID-19 pandemic, it is not encouraged to recruit and gather participants in a controlled lab environment. Hence, we implement a **host mode** to monitor different clients from their perspective and record gaze and video data. The host is a special client that does not participate in the actual study but can act as one of the participants for assisting them with technical issues. By default, the renderer will visualize all participants together with their gaze awareness. The host can observe any client by sending an "observe X" command to the LookAtChat server. The server then returns the layout of all video streams as observed by the designated user. The renderer will visualize the user's gaze information so that the host can verify that the eye-tracking modules are correctly calibrated.

To minimize bandwidth, the host does not send video and audio streams to other clients but only receives the streams from others. For post-study analysis, we record and save all the video, audio, and gaze data from the host machine.

Traditional video conferencing tools may allow the user to privately share their screens with the host. However, our approach reduces unnecessary communication efforts and allows the host to asynchronously examine remote user setups.

Detection of Gaze Awareness. We leverage WebGazer [Papoutsaki et al. 2016] to calibrate and obtain raw gaze positions in each client. Constrained by the webcam set up with an ordinary laptop or PC, WebGazer is state-of-the-art, off-the-shelf software for eye-tracking technologies. Different from WebGazer, which reports gaze coordinates, LookAtChat focuses on "who is looking at whom" for video conferencing. We smooth the coordinate outputs from WebGazer with 1€ Filter [Casiez et al. 2012] and then classify the data to understand which client is being looked at.

Our system expects users to reach an accuracy of 80% during the calibration session and ensure that the size of the face is larger than 25% of the video frame. To improve accuracy, we

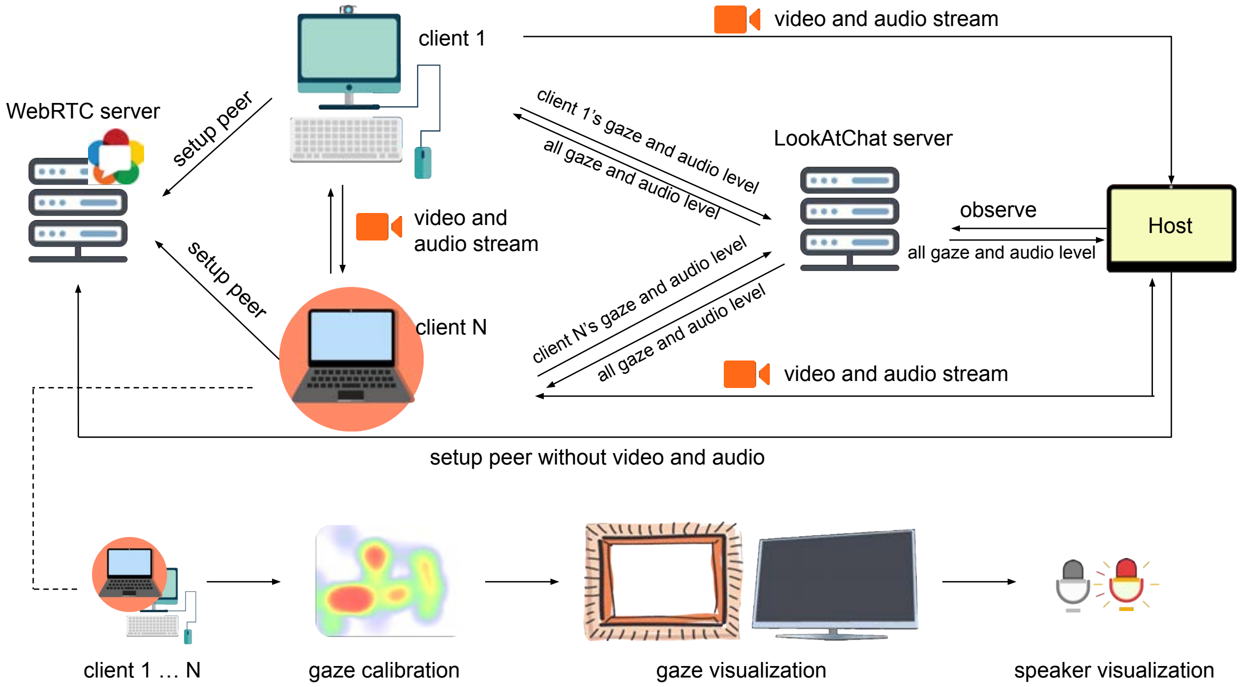


Figure 7.4: LookAtChat workflow. As a regular client, video and audio streams are transmitted to each other in a peer-to-peer manner. Each client is required to calibrate gaze first and then send individual gaze and audio levels to the LookAtChat server after joining the group. The LookAtChat server broadcasts received data to every client. Then each client renders the gaze according to the current layout and the microphone symbol.

fine-tune the parameters of 1€ Filter and video stream placement with a Tobii eye tracker.

First, we tune parameter *mincutoff* in one euro filter to ensure the gaze coordinates are not jittering and *beta* to ensure the result is not introducing too much latency. An animated dot moves from the start to the end of a line. The animated dot is a circle with a 10-pixel radius (from experimental data). We move our cursor to follow the dot several times and record the cluster of cursor positions. The sum of the average distance between the cursor and the animated dot is set as a reference value. We next move our gaze to follow the animated dot and apply the same calculation. *Mincutoff* is tuned to ensure that the distance of gaze is comparable to the reference value. *Beta* is tuned to ensure that the latency is shorter than 5 ms between the raw gaze position and smoothed gaze position on average. We set *mincutoff* to be 0.3 and *beta* to be 0.3 for smoothing the raw gaze positions.

Second, we adjust the placement of the video stream. We calculate a zone ID to specify which video stream is being looked at. For ground truth (data from Tobii), we calculate a valid zone ID when the gaze dot is in the center of a video stream. The size of the central zone is defined as 1/4 of a regular video zone. With smoothed gaze coordinates, we calculate a valid zone ID when it is in the video zone. We reach 95% after changing the distance between two video stream areas horizontally and vertically. Then the proportion of the distance between two video streams (both in x and y) and the entire screen is recorded. In this way, we can ensure that LookAtChat behaves the same on different screens.

Gaze data is constructed as a pair of source ID and destination ID (which could be null) on each client. The data is sent to the LookAtChat server every 16 ms via web sockets to achieve real-time performance.

Rendering. LookAtChat renders two types of data for each client. As we integrate WebRTC into our system for peer-to-peer video communication (including video stream and audio stream), the video stream is rendered as a video texture through Three.js. Additionally, we retrieve the local audio level on each client. The audio level is sent to LookAtChat server and broadcast to all the clients. Thus, each client can see a microphone icon "on" or "off' based on the received audio level data (see "author1" in Figure 7.5(f)). We also record the audio level data for further data analysis.

Gaze data is rendered differently according to the layout. For directional layout, the outglowing effect and the arrows are rendered with increasing opacity. Users can feel the action dynamically from the fade in and out effects. Figure 7.5(a) to (d) show the view of the same user "self" in directional layout. For example, the video stream of user "author1" in (a) is outglowing because user "author1" is looking at user "self". Regarding perspective transformation, the video image of the gaze source is transformed to facing the video image of the target so that users can feel the movement of [who] is tuning and looking at [whom]. If the viewer is being gazed at, the

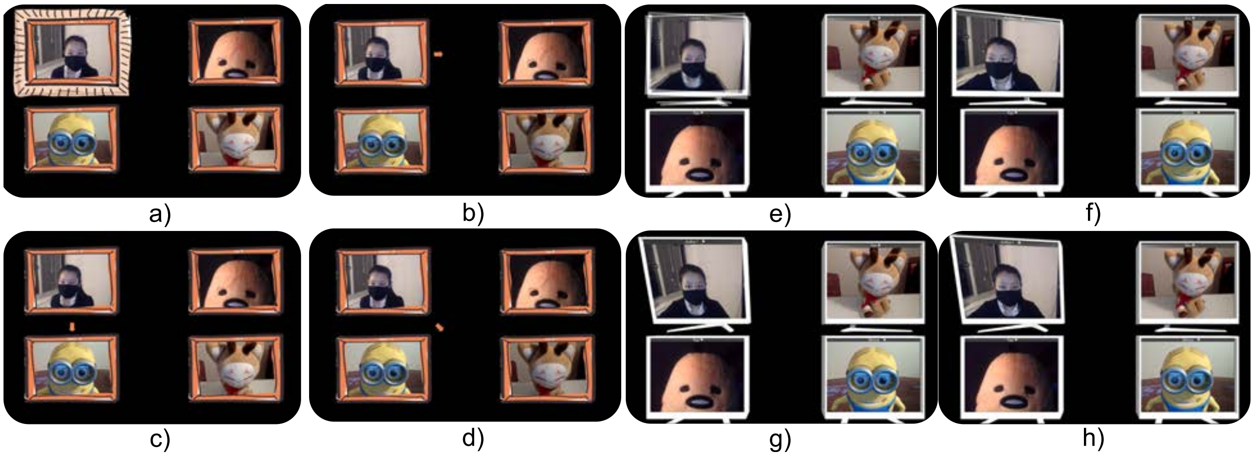


Figure 7.5: a) – d) demonstrate screenshots of how the user recognize mutual gaze (a)) and eye contacts between the person on the upper left and other participants (b) – d)) in the 2D directional layout. Accordingly, e) – h) illustrate the gaze in the 3D perspective layout.

video texture of the gaze source will be slightly shaken. As Figure 7.5(e) illustrates, user "author 1" is looking at "self". Accordingly, user "author1" is looking at "fox" on the right (Figure 7.5(f)), at "egg" underneath (Figure 7.5(g)), and "minions" at the corner (Figure 7.5(h)). Interpolation is applied between different transformations for a smooth experience and also to simulate a "turning" action.

7.4 Evaluation: Who Is the Spy

We conducted a user study to examine how the different layout variants perform in terms of conversation, subjective feedback, and user preferences, compared with a "baseline" layout where no gaze information is visualized. The user study follows a within-subject design in three conditions: baseline, directional layout, and perspective layout. The three conditions were counterbalanced to avoid bias in the following combinations: baseline–directional–perspective, directional–perspective–baseline, perspective–baseline–directional. The DVs were conversation experience defined in Sellen's work [Sellen 1995], user experience defined in Schrepp *et al.*'s work [Schrepp et al. 2017] and Hung *et al.*'s work [Hung and Parsons 2017], and Temple Presence Inventory

(TPI) [Lombard et al. 2009]. We processed the data through an analysis of variance (ANOVA). All tests for significance were made at the $\alpha = 0.05$ level. The error bars in the graphs show the 95% confidence intervals of the means.

7.4.1 Participants and Apparatus

We recruited a total of 20 participants at least 18 years old with normal or corrected-to-normal vision (6 females and 14 males; age range: 24–39, $M = 28.55$, $SD = 3.62$) via social media and email lists. The participants have a diverse background from both academia and industry. None of the participants had been involved with this project before. We assign participants into four 5-person groups for the user study. The study was conducted remotely in personal homes. Participants used their personal computers with a webcam, visited the website we provided through Google Chrome browser, and experienced different conditions as instructed by the host. We instructed participants to take the user study in a quiet and brightly lit room where faces in the webcam are clearly visible from the background. For the duration of the study, participants' behavior, including their conversations, video streams, and gaze positions were observed and recorded.

7.4.2 Procedure

Our remote user study is scheduled using conventional calendar and videoconferencing tools (Zoom). Once all participants were online, the host briefly introduces the LookAtChat system with a tutorial video and asks all participants to fill in consent forms. After the tutorial, the host instructs all participants to enter a designated layout in the user study website (`https://eye.3dvar.com`). Participants are instructed to mute their video&audio streams in Zoom to prevent echoing and save networking bandwidth. Meanwhile, the participants can still follow the host's instructions from Zoom and the host can monitor the experiments with the **host mode** in LookAtChat. The user study session of each condition consists of three parts: gaze calibration

(~5 min), warm-up conversation (~3 min), and two game sessions of "who is the spy" (~20 min). We now describe the three parts in more detail:

Gaze calibration. Participants are required to first calibrate their gaze individually. Our system adopts the calibration procedure of WebGazer[Papoutsaki et al. 2016]: A box rendered around the participant's face mesh turns green when the participant is at the center of the camera view and close enough. Next, the participant calibrates 9 points on the screen and the accuracy of gaze point is reported. We suggest that the participant proceeds after reaching 80%.

Warm-up conversation. At the beginning of each condition, researchers briefly describe how gaze information is visualized for the current condition. Later on, participants pick a topic (self-introduction, favorite TV show, etc.) and one by one give a short speech for around 30 seconds. During the warm-up conversation, participants get familiar with the behavior of the current condition.

Game "who is the spy". After the warm-up conversation, the group is instructed to play a party word game, *"Who is the spy"*, twice. Before the game starts, each participant receives a word: three of them act as detectives and get the same word (*e.g.*, William Shakespeare), while the other two act as spies and get a different word (*e.g.*, Leo Tolstoy). Only the spies know everyone's identities. For each round, each player needs to describe their word, talks about who may be the spies that received a different word. Spies will try to guess detectives' words and pretend they are holding the same word. Detectives will try to describe with ambiguity and infer spies with language and non-verbal cues. This game was selected because it is a conversation-based game that typically requires lots of eye contact to tell who is lying and which two spies are teammates. At the end of each round, each player casts their vote for the spy, and the player with the most votes is put out of the next rounds. At any time, detectives who successfully indict the spies and spies who successfully guess the detectives' word earn points. Each player has only one chance for the indictment or guess.

At the end of each study session, we ask the participants to fill an online questionnaire about

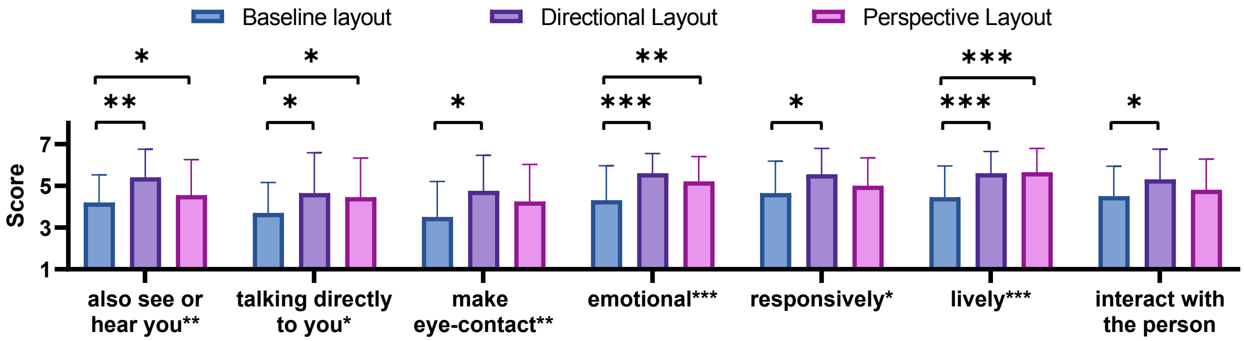


Figure 7.6: Summary of significant results regarding TPI between baseline layout, directional layout, and perspective layout. *: p <= 0.05, **: p <= 0.01, ***: p <= 0.001. 10 questions were selected in the category to be asked. We found 6 of them have significant effects through ANOVA and 7 of them have significant impacts in post hoc tests. Directional layout has notably better scores than baseline layout while perspective layout is slightly better.

conversation experience, user experience, and TPI [Lombard et al. 2009] with a 7-point Likert scale for the condition they just completed (~5 min). Hence, the study session of each condition lasts for around 30 to 45 minutes. At the end of all the three conditions, we ask the participants to fill in demographic information, the scale of usability in general, and rank the conditions. Lastly, participants are interviewed about LookAtChat, reasons for their ranking, and gave suggestions. On average, the experiment takes about 100 to 120 minutes in total.

7.5 Results

We validated that the data satisfies the assumptions of an analysis of variance (ANOVA). All tests for significance were made at the $\alpha = 0.05$ level. The error bars in the graphs show the standard error. Symbol * means $p <= .05$, ** means $p <= .01$, and *** means $p <= .001$.

7.5.1 Social experience

The results for the ratings of social experience questions that have significant results over three layouts are illustrated in Figure 7.6. For the question *"How often did you have the sensation* *that people you saw/heard could also see/hear you?"* ($M_{baseline} = 4.2, M_{dir} = 5.4, M_{per} = 4.55$), a one-way within-subjects ANOVA was conducted to test the influence of layout on the ratings. The main effects for layout ($F(1, 19) = 5.94, p = .006^{**}$) were significant. Post hoc t-tests with Holm correction showed a significant difference between baseline layout and perspective layout ($t(19) = -3.35, p < .006^{**}$) with a 'large' effect size (Cohen's d = .75), as well as between directional layout and perspective layout ($t(19) = 2.37, p < .046^{*}$) with a 'medium' effect size (Cohen's d = .53). The results indicate that LookAtChat with directional layout and perspective layout provided notably more bidirectional sensation than baseline layout.

For the question *"How often did it feel as if someone you saw/heard was talking directly to you?"*($M_{baseline} = 3.7, M_{dir} = 4.65, M_{per} = 4.45$), a one-way within-subjects ANOVA was conducted. The main effects for layout ($F(1, 19) = 4.93, p = .012^{*}$) were significant. Post hoc t-tests with Holm correction showed significant differences between the baseline layout and the other two layouts (both $p < .05^{*}$) with a 'medium' effect size (Cohen's d = .52 to .66). The results suggest that LookAtChat with directional layout and perspective layout provided notably more feelings of direct conversation than baseline layout.

For the question *"How often did you want to or did you make eye-contact with someone you saw/heard?"*($M_{baseline} = 3.5, M_{dir} = 4.75, M_{per} = 4.25$), a one-way within-subjects ANOVA was conducted. The main effects for layout ($F(1, 19) = 6.71, p = .003^{**}$) were significant. Post hoc t-tests with Bonferroni correction showed significant differences between baseline layout and directional layout ($p < .002^{**}$) with a 'large' effect size (Cohen's d > .8). The results indicate that LookAtChat with directional layout provides significantly more eye contact than the baseline layout.

Regarding the social richness questions including emotional ($M_{baseline} = 4.3, M_{dir} = 5.6, M_{per} = 5.2$), responsive ($M_{baseline} = 4.7, M_{dir} = 5.6, M_{per} = 5.0$), and lively ($M_{baseline} = 4.5, M_{dir} = 5.6, M_{per} = 5.7$), a one-way within-subjects ANOVA was conducted for each of them. The main effects for layout were significant on "emotional" (($F(1, 19) = 11.13, p < .001^{**}$)) with post hoc t-tests with Bonferroni correction showed significant differences between baseline layout and the other two layouts (both $p < .006^{**}$) with a ‘large’ effect size (Cohen’s d > .71); on “responsive” ($(F(1, 19) = 3.69, p = .03^{*})$) with post hoc t-tests (holm correction) showed significant difference between baseline layout and directional layout (both $p = .03^{*}$) with a ‘medium’ effect size (Cohen’s d = .6); on “lively” ($(F(1, 19) = 10.65, p < .001^{**})$) and post hoc t-tests with Bonferroni correction showed significant differences between baseline layout and the other two layout (both $p < .001^{***}$) with a ‘large’ effect size (both Cohen’s d > .87). The results indicate that LookAtChat with directional layout and perspective layout provides significantly more social richness feelings than baseline layout.

For other questions discussing social experience, we did not find significant effects over the three layouts. Furthermore, we found that question *“to what extent did you feel you could interact with the person or people you saw/heard?”* has “large” effect size ($\eta^2 = .144$). Post hoc t-tests with Holm correction shows baseline layout has significantly negative effects ($p < .05^{*}$) compared with the directional layout. Hence, the result suggests that LookAtChat with a directional layout brings more interaction potential to users than the baseline layout.

7.5.2 User engagement and experience

The ratings of the question *“The visualization of the layout is clear and balanced.”* over three layouts are illustrated in Figure 7.7 ($M_{baseline} = 5.8, M_{dir} = 5.7, M_{per} = 5.0$). A one-way within-subjects ANOVA was conducted to test the influence of layout on the ratings. The main effects for layout ($F(1, 19) = 3.83, p = .03^{*}$) were significant. Post hoc t-tests with Holm correction showed a significant difference between baseline layout and perspective layout ($t(19) = 2.54, p < .046^{*}$) with a ‘medium’ effect size (Cohen’s d = .57). The results indicate that LookAtChat with baseline and directional layout is more clear and balanced than the perspective layout.

Regarding *“the content or features provided on this website were interesting to me.”* ($M_{baseline} = 4.2, M_{dir} = 5.8, M_{per} = 5.4$), the main effects for layout ($F(1, 19) = 9.15, p = .001^{***}$) was sig-

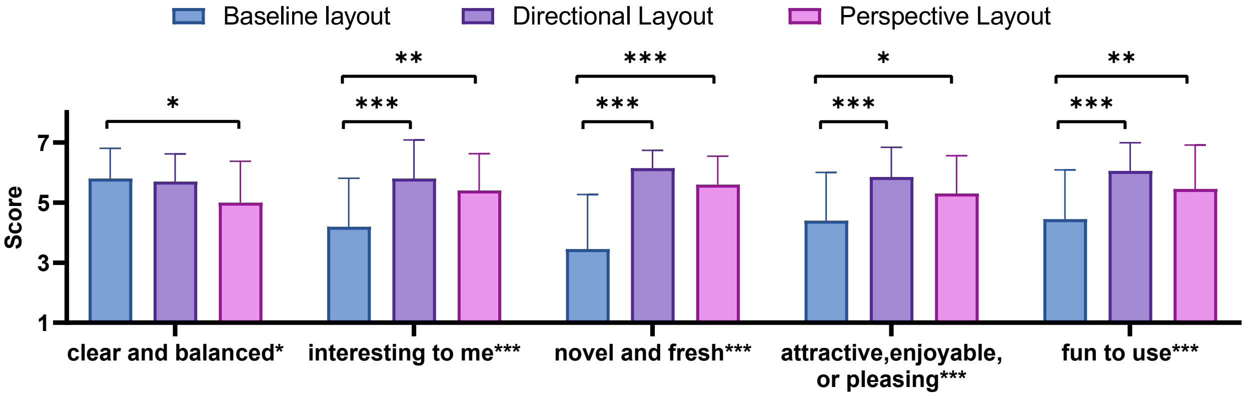


Figure 7.7: Summary of significant results regarding user engagement and user experience between baseline layout, directional layout, and perspective layout. *: p <= 0.05, **: p <= 0.01, ***: p <= 0.001. We provided 6 questions on user engagement and 4 questions on user experience for participants to fill. Results show that 3 user engagement questions and 2 user experience questions have significant effects through ANOVA. Directional layout and perspective layout both have significant effects on participants' feedback of feeling "interesting", "novel", "attractive" and "fun".

nificant with a 'large' effect size ($\omega^2 = 0.18$). Post hoc t-tests with Holm correction showed a significant difference between baseline layout and directional layout ($t(19) = -4.11, p < .001^{***}$) and perspective layout ($t(19) = -3.08, p < .008^{**}$). The results indicate that LookAtChat with baseline layout is significantly less interesting than directional and perspective layout.

In terms of *"the features provided by this website were novel and fresh."* ($M_{baseline} = 3.45, M_{dir} = 6.15, M_{per} = 5.6$), the main effects for layout ($F(1, 19) = 32.76, p < .001^{***}$) were significant with a 'large' effect size ($\omega^2 = 0.47$). Post hoc t-tests with Holm correction showed a significant difference between baseline layout and directional layout and perspective layout (both$p < .001^{***}$). The results indicate that LookAtChat with baseline layout is significantly less novel or fresh than directional and perspective layout.

Speaking of the overall impression of the design (attractive, enjoyable, or pleasing) with $M_{baseline} = 4.4, M_{dir} = 5.9, M_{per} = 5.3$. The main effects for layout ($F(1, 19) = 9.28, p = .001^{***}$) were significant with a 'large' effect size ($\omega^2 = 0.16$). Post hoc t-tests with Holm correction showed a significant difference between baseline layout and directional layout ($t(19) = -4.3, p < .001^{***}$) and perspective layout ($t(19) = -2.6, p = .02^{*}$). The result for ratings of *"fun to use"*

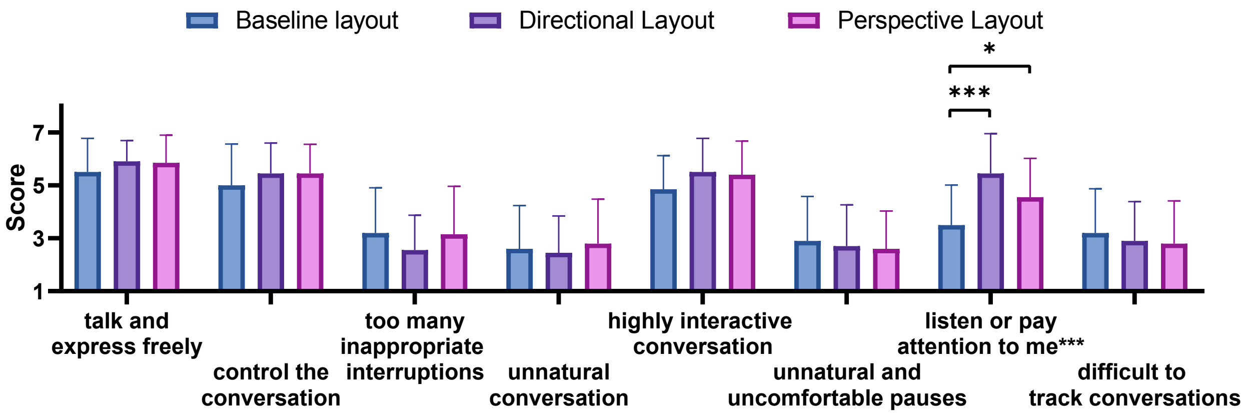


Figure 7.8: Summary of the results regarding conversation experience between baseline layout, directional layout, and perspective layout. *: p <= 0.05, **: p <= 0.01, ***: p <= 0.001. Directional layout and perspective layout both have significant effects on the feeling of "attention". For other questions, directional layout and perspective layout have better but not significant scores than baseline layout.

over three layouts are $M_{baseline} = 4.5, M_{dir} = 6.1, M_{per} = 5.5$. The main effects for layout ($F(1, 19) = 11.49, p = .001^{***}$) were significant with a 'large' effect size ($\omega^2 = 0.17$). Post hoc t-tests with Holm correction showed a significant difference between baseline layout and directional layout ($t(19) = -4.75, p < .001^{***}$) and perspective layout ($t(19) = -2.97, p = .01^{**}$). The results suggest that LookAtChat with the baseline layout is significantly less attractive and fun than the directional and the perspective layouts overall.

7.5.3 Conversation experience

The results for the ratings of question *"I knew when people were listening or paying attention to me."* over three layouts are illustrated in Figure 7.8 ($M_{baseline} = 3.5, M_{dir} = 5.45, M_{per} = 4.55$). A one-way within-subjects ANOVA was conducted to test the influence of layout on the ratings. The main effects for layout ($F(1, 19) = 12.50, p = .001^{***}$) were significant. Post hoc t-tests with Holm correction showed a significant difference between baseline layout and directional layout ($t(19) = -5.00, p < .001^{***}$) and perspective layout ($t(19) = -2.68, p < .02^{*}$) with a 'large' effect size (both Cohen's d > .6). The results indicate that LookAtChat with both directional layout and perspective layout provided notably more feedback of attention than the baseline layout.

We found there was no significant effect of layout on other conversation questions. Post hoc t-tests with Holm correction did not show significant differences among all questions in this category.

7.5.4 Usability, Preferences and Subjective feedback

Participants rated the usability of LookAtChat in general after they experienced all three layouts. Participants (like P10,M) thought the system is "*easy to understand*". The overall usability score is 72.5, which is interpreted to have higher perceived usability than 70% of all products tested.

Fourteen participants preferred the directional layout, five participants preferred the perspective layout, and one preferred the baseline layout. P6(M) thought "*arrows and highlights are easier for me to notice the other's attention*". Similarly, P11(F) agreed that "*the highlight of when people look at you is more obvious.*". For participants who preferred perspective layout, participants found it "*fun*"(P9, M) and "*having such eye contact design in video conferencing is interesting and I feel the motive of being focus on other's speeches.*" (P7, F). Additionally, P1(M) and P2(F) shared similar comments that "*more intuitive than the other designs*" and "*simple form but it also gives you info*". The only user (P4, F) who preferred the baseline layout explained that the baseline layout is "*succinct*". From the reason of preferences reported by the participants, we found that participants tended to choose the design that is "clear and easy to understand" (P18, F) and (P12, F), however, they have different perceptions on defining "clear". Providing information that is more than users expect leads to negative results. An automatic adjuster or manual one would be useful.

During the interview, we asked all the participants whether they think gaze awareness is worth visualizing as well as spatial relationship. 15 out of 20 agreed it is helpful and others have concerns about being overwhelmed. We further asked how they think about gaze awareness between themselves and other people, and eye contact between other people. 19 out of 20 reported they want to know if they are being looked at as well as whether other people receive their gaze information. We believe that visualizing gaze awareness data will improved users' engagement, social richness, and helpful to the conversation experience while videoconferencing. Since LookAtChat only requires a consumable webcam, participants such as P4(F) and P5(M) reported that it should be feasible to be integrated into existing videoconferencing systems, although may cause extra rendering costs.

7.6 Discussion and Conclusion

In general, LookAtChat is an *"easy-to-use"* system with an overall usability score of 72.5 in this user study. Participants found it *"easy to understand"* and *"fun to use"*. Compared with the baseline layout, the directional layout and the perspective layout provide better feedback on social experience, user engagement, and conversation experience.

RQ1: How do people perceive eye contact without visualization in conventional videoconferencing?

Informed by the interview feedback, participants usually don't interpret gaze awareness in videoconferencing. Commented by P18(F), "*I focus more on the audio so I won't miss what other people are talking about.*". In the meantime, participants are used to interpreting gaze offset to a very high degree of accuracy [Grayson and Monk 2003]. Hence, providing gaze awareness through the user interface is worth researching in parallel with gaze correction technologies.

RQ2: Can visualization of eye contact improve communication efficiency?

According to the conversation experience feedback, participants have better experiences with conversation flow, including self-expression, controlling the conversation, and tracking the conversation, but not significantly. Participants felt the conversation is more interactive. Meanwhile, participants reported fewer inappropriate interruptions, fewer uncomfortable pauses, in addition to feeling less unnatural when using LookAtChat with directional or perspective layouts though not significantly. Importantly, LookAtChat has a significant effect on participants' belief that other people are listening and paying attention to them. As P4(F) reported, "*visualization of eye contact is helpful, knowing that people were watching at least helped me stay focused for the entire time.*"

RQ3: In what circumstances will users prefer to see eye contact?

We found that participants have different preferences while in different roles, as a meeting host, or as a talk attender. Also, the purpose of the video conference mattered. As P5(M) described, "business meetings may value more on the quality of video image however colorful UI designs (like arrows in directional layout) may distract people from that". For small-group discussion, P4(F) agreed that "providing such information motivates me to engage more in such video conferencing like brainstorming." Furthermore, participants placed more value on gaze awareness relevant to themselves than between other participants. P5(M) thought gaze awareness between other participants "*are helpful but too much for me to process at one time*". Investigating the effects of enabling eye contact among other participants is worth researching.

RQ4: What forms of eye contact visualization may be preferred by users?

More participants preferred the directional layout (N=14) than the perspective layout (N=5). The directional layout shows arrows and an outer glow that "*are easier to notice other's attention*"(P6, M). It applies "*no change on video image*"(P18, F) and indicates the interaction "*more visually for better connection*" (P13, M). The perspective layout provides spatial relationships in 3D. As reported by P1(M), it has "*better visual presentation of gazing at somebody, more intuitive*". P7(F) Comparing directional and perspective layout, "*directional layout is somehow too obvious and may require extra effort to focus on my speech*" while "*perspective layout eliminates this issue and I got the balance between freely speaking and knowing I was listened to by others.*". Briefly, we can tell that participants prefer the designs that are "clear" to them and with a smooth transition between looking at self and others.

Design Implications:

1. Show the gaze awareness intuitively.

In video conferences, participants mainly focus on the conversation. If the information provided through the visual design is too indirect and may require additional cognitive load, participants may feel distracted and lost in the conversation. For example, out-glowing in the directional layout receives positive comments from 7 participants because it is easy to understand.

2. Control the visualization level.

   Participants' preference is affected by “how attractive the design is” and “how I want the design to be attractive”. For example, P7(F) preferred the perspective layout because the directional layout is relatively more distracting to her when she wanted to focus on listening. Providing a slider for adjusting the level and automatically controlling the level with audio data is helpful.

3. Provide the control of gaze data transmission.

   As video conferencing users have the option to mute or hide the video during video conferencing, most participants want to hide or send anonymous gaze data to the host in video conferences.

4. Scale the design for various user scenarios.

   Although we only evaluate small-group setup for LookAtChat, we designed and interviewed participants about their opinions on large team meetings or presentations with a large audience. An important future direction is to adapt the system to fit different use cases and larger numbers of users.

5. Provide host mode.

   Host mode is not only helpful for researchers to understand participants from their views but essential for conducting remote user studies as well.

Limitations. While LookAtChat is designed for remote video conferencing, as a proof-of-concept, we do not support more than one participant to be co-located. Our eye-contact detection algorithm only supports one user in front of the webcam and the accuracy is limited by the algorithms and individual calibration procedures. In terms of the user study, we only evaluate small-group discussions without shared presentations. As the ages of our participants spanned 24 - 39, the results of our study may not generalize to other populations such as junior students or elder adults who may prefer more or fewer eye contacts in video conferences. As our user study was conducted remotely, the bandwidth of home networks may impact how the users perceive our system. A small number of participants encountered frame-dropping during the conversation due to networking issues, which may negatively impact their assessment. Furthermore, as users of our system were only able to engage with LookAtChat for casual and gaming conversations, their assessment of how such a system may help in other use cases such as team meetings or lectures is not fully conclusive.

7.6.1 Conclusion

In this chapter, we introduce LookAtChat, a web-based video conferencing system that supports visualizing eye contact for small-group conversations. Motivated by missing gaze information in conventional video conferences, we investigate the demands of gaze information by conducting five formative interviews. We further explore the design space of visualizing eye contacts with video streams of small groups and propose 11 layouts by brainstorming in focused groups. As a proof-of-concept, we develop LookAtChat which supports eye contact visualization for small-group conversations. We conduct a remote user study of 20 participants to examine the benefits and limitations of the interfaces, as well as the potential impacts of user engagement and experience on the conversation. The quantitative results indicate that LookAtChat with directional layout and perspective layout provided notably more bidirectional sensation, feelings of direct conversation, social experience, and engagement than the baseline layout. More participants prefer the 2D directional layout to the 3D perspective layout because it is simpler and easier to understand.

We plan to explore several future directions for improving LookAtChat. First, we plan to implement more layouts from our design space exploration stage and establish an open-source community to develop more layouts for the system. Second, we intend to integrate privacy protection filters for users to select whether or not to share their gaze information. Third, more advanced real-time neural models may be leveraged to improve the tracking accuracy in LookAtChat and balance the trade-off between accuracy and real-time performance.

As an initial step toward visualizing gaze awareness in conventional video conferencing interfaces with commonly accessible hardware requirements, we believe our work may inspire more designs to convey nonverbal cues for remote conversations. Such features may eventually be integrated with video conferencing software to increase social engagement and improve the conversation experience.

Table 7.1: Proposed visualization of eye contact for remote video conferences

Category	Name	Description
2D flat layout	directional layout	Depict arrows between video streams to indicate sources and targets of eye contacts, while out-glowing the video window of users who are looking at the current user.
	animated flows	Render dynamic flows instead of arrows to convey eye contacts; the sizes of the flows are proportional to the duration of the gaze actions.
	text overlay	Overlay the text of "[who] is looking at [whom]" directly on captions of the video window.
	color highlights	Change the color of the video border to indicate the eye contacts while each distinct color is assigned to each participant.
	icon overlay	Append users' profile pictures to the caption area of the video window to convey eye contact.
3D immersed layout	perspective layout	Apply perspective transformation of the video window to imply eye contact between other participants and gently shake the video of users who are looking at the current user.
	avatar / top-view	Render a top-view of 3D avatars of all users alongside with their video streams and change their rotation according to gaze actions.
	avatar / first-person	Warp live video streams to the 3D avatars of all users positioned along a curve; rotate the avatars to reflect their gaze actions; present a first-person perspective for the current user
	avatar / third-person	Based on "avatar / first-person", present a third-person perspective with isometric projection[Cai et al. 2007].
hybrid layout	split-view	Present a 2D flat layout and a 3D immersed layout side by side. Hence the 3D avatar layout doesn't need to wrap video streams to the avatars to avoid "uncanny valley" effects.
	picture-in-picture	Depict a 2D flat layout with video windows in full while a 3D avatar layout is rendered as an overview thumbnail at the screen center to convey eye contacts.

8 Conclusion and Future Work

In this final chapter, I conclude by summarizing the major contributions and highlighting promising avenues for future exploration.

8.1 Summary of Contributions

8.1.1 PhyShare

I propose a new approach for interaction in virtual reality via robotic haptic proxies, specifically targeted towards collaborative experiences, both remote and local. I present several prototypes utilizing our three mapping definitions, demonstrating that robotic proxies can be temporarily assigned to represent different virtual objects, that PhyShare can allow remotely located users to have the experience of touching on the same virtual object, and that users can alternately use gestures to command objects without touching them.

8.1.2 CollaboVR

CollaboVR is an end-to-end collaboration system using a cloud-based computing architecture to support multi-user sketch, audio communication, and collaboration in 3D. It leverages real-time techniques to share freehand sketches, convert 2D sketches into 3D models, and interact with animations in collaborative Virtual Reality. Custom configurations are designed for real-time user arrangements and input modes for multi-user sketching scenarios inspired by real-world metaphors. Two scenarios: teaching and brainstorming are evaluated and guidelines for immersive presentation design are concluded.

8.1.3 TapGazer

TapGazer is a novel text entry method combining tapping and gaze. TapGazer was designed to facilitate text entry while wearing VR/AR HMDs, without the need for a physical keyboard or to look at one's hands, in anticipation of a future where affordable AR glasses will be as ubiquitous as mobile phones are today. TapGazer makes several key contributions: a design that combines tap and gaze for effective text entry, with variants for use without gaze tracking and for accommodating different user preferences. A simulated study showing that TapGazer is usable, easy-to-learn for QWERTY users, and able to reach average speeds of 52.5 wpm, 77% of their QWERTY typing speed when using the GC variant. A model-based performance analysis illustrating the effects of different design options and usage strategies.

8.1.4 LookAtChat

LookAtChat is a web-based video conferencing system that supports visualizing eye contacts for small-group conversations. Motivated by missing gaze information in conventional video conferences, I investigate the demands of gaze information by conducting five formative interviews. Then I enumerate design implications through formative interviews and extending the design space of visualizing gaze and spatial information in video conferences. Followed by reporting evaluation results and reflections about the opportunistic use of eye contact visualization in video conferencing systems - benefits, limitations, and potential impacts on future remote collaboration systems. To facilitate future development in video conferencing systems with visualization of nonverbal cues, A live demo is available at `https://eye.3dvar.com`.

8.2 Future Work

Each piece of work that is completed opens up new research possibilities and questions. The field of on-world computing is vast, and the tools that I have built to explore the space will be of great help in navigating the work that is yet to come. In this section, I highlight and describe some of the most promising unanswered research questions which have the potential to further our understanding and our capabilities in enhancing collaboration and productivity for VR and AR.

8.2.1 Improvements to Haptic Feedback

First of all, I evaluate only three physical mappings, omitting alternate versions. For example, physical mapping can also be applied to a fixed environment, or even incur even human as one of the proxies. Our mapping only emphasizes the mapping between robots and virtual objects without considering other stakeholders.

Apart from that, I did not explore any moving algorithms for robots related to our proposed mappings. While our work mainly focuses on the scenario, design, consideration, and outcomes between the relationships of physical objects and their virtual representation, we believe that moving algorithms will play a more vital role when the mapping is non-linear. For example, Sun et. al. proposed a nonlinear moving algorithm of users when a floor plan is provided [Sun et al. 2016]. A moving algorithm that effectively distorts the virtual environment for minimizing physical movement will be our next direction of work.

8.2.2 Improvements to Collaborative VR

I only explore very limited configurations in CollaboVR. It is beneficial to extend the design space of sketch-based interaction, explore the effects of non-verbal cues in multi-user communication, and add deep-learning-based models as cloud-hosted applications in CollaboVR. There is more to explore for further manipulation on the content like enabling surface and screen display together and the participant representation which could be symmetric or asymmetric.

8.2.3 Improvement to Productivity in VR

There are other alternatives for input device design. How to seamlessly connect the interaction of text entry and other functionality in VR is worth researching. Adding rich text edit capabilities will improve the user's productivity in VR. Furthermore, manipulating documents and other digital information via immersive environments is worth discussion too.

8.3 Conclusion

The immersive environment is now popular and consumable, but the experiences are confined to a lack of haptic feedback, collaborative support, and communication enhancement, as well as an efficient text entry method for facilitating productivity in VR. This thesis initiates potential solutions to enhance collaboration and productivity for immersive environments.

Bibliography

Adhikary, J. (2018). Text entry in vr and introducing speech and gestures in vr text entry. In *MobileHCI*, pages 1083–1092, Barcelona, Spain. Association for Computing Machinery.

Ahn, S. and Lee, G. (2019). Gaze-assisted typing for smart glasses. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 857–869.

Albolino, S., Cook, R., and O'Connor, M. (2007). Sensemaking, Safety, and Cooperative Work in the Intensive Care Unit. *Cognition, Technology & Work*, 9(3):131–137.

Andersson, R. L. and Chen, H. H. (1997). Method for Achieving Eye-to-Eye Contact in a Video-Conferencing System. US Patent 5,675,376.

Andersson, R. L., Chen, T., and Haskell, B. G. (1996). Video Conference System and Method of Providing Parallax Correction and a Sense of Presence. US Patent 5,500,671.

Araujo, B., Jota, R., Perumal, V., Yao, J. X., Singh, K., and Wigdor, D. (2016). Snake charmer: Physically enabling virtual objects. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 218–226. ACM.

Ashdown, M., Oka, K., and Sato, Y. (2005). Combining Head Tracking and Mouse Input for a GUI on Multiple Monitors. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pages 1188–1191.

Azmandian, M., Hancock, M., Benko, H., Ofek, E., and Wilson, A. D. (2016). Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1968–1979. ACM.

Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594.

Beck, S., Kunert, A., Kulik, A., and Froehlich, B. (2013). Immersive Group-to-Group Telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):616–625.

Bee, N. and André, E. (2008). Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *International Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems*, pages 111–122. Springer.

Benko, H., Holz, C., Sinclair, M., and Ofek, E. (2016). Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728. ACM.

Benko, H., Wilson, A. D., and Zannier, F. (2014). Dyadic Projected Spatial Augmented Reality. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 645–655. ACM.

Benligiray, B., Topal, C., and Akinlar, C. (2019). Slicetype: fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces*, 13(4):321–334.

Bergig, O., Hagbi, N., El-Sana, J., and Billinghurst, M. (2009). In-Place 3D Sketching for Authoring and Augmenting Mechanical Systems. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 87–94. IEEE.

Bi, X., Smith, B. A., and Zhai, S. (2010). Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 283–286.

Billinghurst, M. and Kato, H. (2000). Out and About—real World Teleconferencing. *BT Technology Journal*, 18(1):80–82.

Billman, D. and Bier, E. A. (2007). Medical Sensemaking With Entity Workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 229–232. ACM.

Blumrosen, G., Sakuma, K., Rice, J. J., and Knickerbocker, J. (2020). Back to finger-writing: Fingertip writing technology based on pressure sensing. *IEEE Access*, 8:35455–35468.

Boletsis, C. and Kongsvik, S. (2019). Controller-based text-input techniques for virtual reality: An empirical comparison. *International Journal of Virtual Reality*, 19(3):2–15.

Bork, F., Barmaki, R., Eck, U., Yu, K., Sandor, C., and Navab, N. (2017). Empirical Study of Non-Reversing Magic Mirrors for Augmented Reality Anatomy Learning. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 169–176. IEEE.

Bovet, S., Kehoe, A., Crowley, K., Curran, N., Gutierrez, M., Meisser, M., Sullivan, D. O., and Rouvinez, T. (2018). Using traditional keyboards in vr: Steamvr developer kit and pilot game user study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–9. IEEE.

Bowman, D., Ly, V., Campbell, J., and Tech, V. (2001). Pinch keyboard: Natural text input for immersive virtual environments.

Brave, S., Ishii, H., and Dahley, A. (1998). Tangible interfaces for remote collaboration and communication. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 169–178. ACM.

Brooke, J. et al. (1996). SUS-A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, 189(194):4–7.

Brun, D., Gouin-Vallerand, C., and George, S. (2019). Keycube is a kind of keyboard (k3). In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–4.

Butscher, S., Hubenschmid, S., Müller, J., Fuchs, J., and Reiterer, H. (2018). Clusters, Trends, and Outliers: How Immersive Technologies Can Facilitate the Collaborative Analysis of Multidimensional Data. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Cai, D., He, X., Han, J., et al. (2007). Isometric Projection. In *AAAI*, pages 528–533.

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410.

Casiez, G., Roussel, N., and Vogel, D. (2012). 1 euro filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530.

Castellucci, S. J., MacKenzie, I. S., Misra, M., Pandey, L., and Arif, A. S. (2019). Tiltwriter: design and evaluation of a no-touch tilt-based text entry method for handheld devices. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–8.

Cavallo, M., Dholakia, M., Havlena, M., Ocheltree, K., and Podlaseck, M. (2019). Dataspace: a Reconfigurable Hybrid Reality Environment for Collaborative Information Analysis. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 145–153. IEEE.

Cheat, M. and Wongsaisuwan, M. (2018). Eye-swipe typing using integration of dwell-time and dwell-free method. In *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 205–208. IEEE.

Chen, S., Wang, J., Guerra, S., Mittal, N., and Prakkamakul, S. (2019). Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6.

Cheng, L.-P., Lühne, P., Lopes, P., Sterz, C., and Baudisch, P. (2014). Haptic turk: a motion platform based on people. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3463–3472. ACM.

Cheng, L.-P., Marwecki, S., and Baudisch, P. (2017). Mutual human actuation. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 797–805. ACM.

Cheng, L.-P., Roumen, T., Rantzsch, H., Köhler, S., Schmidt, P., Kovacs, R., Jasper, J., Kemper, J., and Baudisch, P. (2015). Turkdeck: Physical virtual reality based on people. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 417–426. ACM.

Chin, J. P., Diehl, V. A., and Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–218. ACM.

Choi, I. and Follmer, S. (2016). Wolverine: A wearable haptic interface for grasping in vr. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 117–119. ACM.

Chow, K., Coyiuto, C., Nguyen, C., and Yoon, D. (2019). Challenges and Design Considerations for Multimodal Asynchronous Collaboration in VR. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–24.

Chung, C.-W., Lee, C.-C., and Liu, C.-C. (2013). Investigating Face-to-Face Peer Interaction Patterns in a Collaborative Web Discovery Task: the Benefits of a Shared Display. *Journal of Computer Assisted Learning*, 29(2):188–206.

Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1):155.

Costagliola, G., Fuccella, V., and Di Capua, M. (2011). Text entry with keyscretch. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 277–286.

Criminisi, A., Shotton, J., Blake, A., and Torr, P. H. (2003). Gaze Manipulation for One-to-One Teleconferencing. In *ICCV*, volume 3, pages 13–16.

Davis, N. M. (2013). Human-Computer Co-Creativity: Blending Human and Computational Creativity. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Dervin, B. (1992). From the Mind's Eye of the User: the Sense-Making Qualitative-Quantitative Methodology. *Sense-Making Methodology Reader*.

Dijksman, J. A. and Khan, S. (2011). Khan Academy: the World's Free Virtual School. In *APS Meeting Abstracts*.

Du, R., Li, D., and Varshney, A. (2019). Geollery: a Mixed Reality Social Media Platform. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 685. ACM.

Du, R. and Varshney, A. (2016). Social Street View: Blending Immersive Street Views With Geo-Tagged Social Media. In *Proceedings of the 21st International Conference on Web3D Technology*, Web3D, pages 77–85. ACM.

Dube, T. J. and Arif, A. S. (2019). Text entry in virtual reality: A comprehensive review of the literature. In *International Conference on Human-Computer Interaction*, pages 419–437. Springer.

Dube, T. J. and Arif, A. S. (2020). Impact of key shape and dimension on text entry in virtual reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Dudley, J., Benko, H., Wigdor, D., and Kristensson, P. O. (2019). Performance envelopes of virtual keyboard text input strategies in virtual reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 289–300. IEEE.

Dudley, J. J., Vertanen, K., and Kristensson, P. O. (2018). Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(6):1–40.

Dunlop, M. D., Durga, N., Motaparti, S., Dona, P., and Medapuram, V. (2012). Qwerth: an optimized semi-ambiguous keyboard design. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion*, pages 23–28.

Elvezio, C., Sukan, M., Oda, O., Feiner, S., and Tversky, B. (2017). Remote Collaboration in AR and VR Using Virtual Replicas. In *ACM SIGGRAPH 2017 VR Village*, pages 1–2.

Eng, W. Y., Min, D., Nguyen, V.-A., Lu, J., and Do, M. N. (2013). Gaze Correction for 3D Tele-Immersive Communication System. In *IVMSP 2013*, pages 1–4. IEEE, IEEE.

Evans, F., Skiena, S., and Varshney, A. (1999). Vtype: Entering text in a virtual world. *submitted to International Journal of Human-Computer Studies*.

Everitt, K. M., Klemmer, S. R., Lee, R., and Landay, J. A. (2003). Two worlds apart: bridging the gap between physical and virtual media for distributed design collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 553–560. ACM.

Fashimpaur, J., Kin, K., and Longest, M. (2020). Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–7.

Findlay, J. M. (1997). Saccade target selection during visual search. *Vision research*, 37(5):617–631.

Follmer, S., Leithinger, D., Olwal, A., Hogge, A., and Ishii, H. (2013). inform: dynamic physical affordances and constraints through shape and object actuation. In *Uist*, volume 13, pages 417–426.

Franz, J., Alnusayri, M., Malloch, J., and Reilly, D. (2019). A Comparative Evaluation of Techniques for Sharing AR Experiences in Museums. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–20.

Fried, O., Tewari, A., Zollhöfer, M., Finkelstein, A., Shechtman, E., Goldman, D. B., Genova, K., Jin, Z., Theobalt, C., and Agrawala, M. (2019). Text-Based Editing of Talking-Head Video. *ACM Transactions on Graphics (TOG)*, 38(4):1–14.

Fuhrmann, A. L., Prikryl, J., Tobler, R. F., and Purgathofer, W. (2001). Interactive content for presentations in virtual reality. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '01, pages 183–189, New York, NY, USA. ACM.

Gauglitz, S., Nuernberger, B., Turk, M., and Höllerer, T. (2014). World-stabilized annotations and virtual scene navigation for remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 449–459. ACM.

Gemmell, J., Toyama, K., Zitnick, C. L., Kang, T., and Seitz, S. (2000). Gaze Awareness for Video-Conferencing: a Software Approach. *IEEE MultiMedia*, 7(4):26–35.

Genest, A. M., Gutwin, C., Tang, A., Kalyn, M., and Ivkovic, Z. (2013). KinectArms: a Toolkit for Capturing and Displaying Arm Embodiments in Distributed Tabletop Groupware. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 157–166. ACM.

Gizatdinova, Y., Špakov, O., and Surakka, V. (2012). Comparison of video-based pointing and selection techniques for hands-free text entry. In *Proceedings of the international working conference on advanced visual interfaces*, pages 132–139.

Gong, J., Xu, Z., Guo, Q., Seyed, T., Chen, X., Bi, X., and Yang, X.-D. (2018). Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Goodman, J., Venolia, G., Steury, K., and Parker, C. (2002). Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 194–195.

Google (2019). AutoDraw Is a New Kind of Drawing Tool That Pairs the Magic of Machine Learning With Drawings From Talented Artists to Help Everyone Create Anything Visual, Fast. https://www.autodraw.com/.

Grandi, J. G., Debarba, H. G., Bemdt, I., Nedel, L., and Maciel, A. (2018). Design and Assessment of a Collaborative 3D Interaction Technique for Handheld Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 49–56. IEEE.

Grandi, J. G., Debarba, H. G., and Maciel, A. (2019). Characterizing Asymmetric Collaborative Interactions in Virtual and Augmented Realities. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 127–135. IEEE.

Grayson, D. M. and Monk, A. F. (2003). Are You Looking at Me? Eye Contact and Desktop Video Conferencing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(3):221–243.

Green, N., Kruger, J., Faldu, C., and St. Amant, R. (2004). A reduced qwerty keyboard for mobile text entry. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1429–1432.

Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., and Kristensson, P. O. (2018). Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 159–166. IEEE.

Gugenheimer, J., Dobbelstein, D., Winkler, C., Haas, G., and Rukzio, E. (2016). Facetouch: Enabling touch interaction in display fixed uis for mobile virtual reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 49–60.

Gugenheimer, J., Stemasov, E., Frommel, J., and Rukzio, E. (2017). ShareVR: Enabling Co-located Experiences for Virtual Reality Between Hmd and Non-Hmd Users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4021–4033. ACM.

Gumienny, R., Gericke, L., Quasthoff, M., Willems, C., and Meinel, C. (2011). Tele-Board: Enabling Efficient Collaboration in Digital Design Spaces. In *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 47–54. IEEE.

Gupta, A., Ji, C., Yeo, H.-S., Quigley, A., and Vogel, D. (2019). Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Gutwin, C. and Greenberg, S. (1996). Workspace Awareness for Groupware. In *Conference Companion on Human Factors in Computing Systems*, pages 208–209. ACM.

Gutwin, C. and Greenberg, S. (1998). Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness.

Gutwin, C. and Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work (CSCW)*, 11(3-4):411–446.

Han, S., Liu, B., Wang, R., Ye, Y., Twigg, C. D., and Kin, K. (2018). Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG)*, 37(4):1–10.

Harley, D., Verni, A., Willis, M., Ng, A., Bozzo, L., and Mazalek, A. (2018). Sensory VR: Smelling, Touching, and Eating Virtual Reality. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 386–397.

Harrison, B. L., Ishii, H., Vicente, K. J., and Buxton, W. A. (1995). Transparent Layered User Interfaces: an Evaluation of a Display Design to Enhance Focused and Divided Attention. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 317–324.

Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA.

He, Z., Zhu, F., and Perlin, K. (2017). Physhare: Sharing physical interaction in virtual reality. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 17–19, New York, NY, USA. ACM.

He, Z., Zhu, F., Perlin, K., and Ma, X. (2018). Manifest the invisible: Design for situational awareness of physical environments in virtual reality. *Arxiv Preprint Arxiv:1809.05837*.

Heo, H., Park, H. K., Kim, S., Chung, J., Lee, G., and Lee, W. (2014). Transwall: A transparent double-sided touch display facilitating co-located face-to-face interactions. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pages 435–438, New York, NY, USA. ACM.

Heun, V., Hobin, J., and Maes, P. (2013). Reality Editor: Programming Smarter Objects. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pages 307–310.

Hickmann, M. and Robert, S. (2006). *Space in languages: Linguistic systems and cognitive categories*, volume 66. John Benjamins Publishing.

Hoyer, H., Jochheim, A., ROhrig, C., and Bischoff, A. (2004). A multiuser virtual-reality environment for a tele-operated laboratory. *IEEE Transactions on education*, 47(1):121–126.

Hsu, T.-W., Tsai, M.-H., Babu, S. V., Hsu, P.-H., Chang, H.-M., Lin, W.-C., and Chuang, J.-H. (2020). Design and Initial Evaluation of a VR Based Immersive and Interactive Architectural Design Discussion System. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 363–371. IEEE.

Huckauf, A. and Urbina, M. H. (2008). Gazing with peyes: towards a universal input for various applications. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 51–54.

Hung, Y.-H. and Parsons, P. (2017). Assessing User Engagement in Information Visualization. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1708–1717.

Huo, K., Wang, T., Paredes, L., Villanueva, A. M., Cao, Y., and Ramani, K. (2018). SynchronizAR: Instant Synchronization for Spontaneous and Spatial Collaborations in Augmented Reality. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pages 19–30. ACM.

Inc., N. (2016). Swype.

Interactive, A. (2015). Real virtuality. http://artaniminteractive.com/real-virtuality/.

Irlitti, A., Piumsomboon, T., Jackson, D., and Thomas, B. H. (2019). Conveying Spatial Awareness Cues in XR Collaborations. *IEEE Transactions on Visualization and Computer Graphics*, 25(11):3178–3189.

Isenberg, P., Fisher, D., Morris, M. R., Inkpen, K., and Czerwinski, M. (2012). Co-located Collaborative Visual Analytics around a Tabletop Display. In *IEEE Transactions on Visualization and Computer Graphics*, volume 18, pages 689–702. IEEE.

Ishii, H. (2008). Tangible bits: beyond pixels. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages xv–xxv. ACM.

Ishii, H. and Kobayashi, M. (1992). ClearBoard: a Seamless Medium for Shared Drawing and Conversation With Eye Contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 525–532. ACM.

Ishii, H., Kobayashi, M., and Grudin, J. (1993). Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments. *ACM Transactions on Information Systems (TOIS)*, 11(4):349–375.

Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241. ACM.

Iwata, H., Yano, H., Fukushima, H., and Noma, H. (2005). Circulafloor [locomotion interface]. *IEEE Computer Graphics and Applications*, 25(1):64–67.

Jacob, R. J. (1993). Eye movement-based human-computer interaction techniques: Toward non-command interfaces. *Advances in human-computer interaction*, 4:151–190.

Jiang, H. and Weng, D. (2020). Hipad: Text entry for head-mounted displays using circular touchpad. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 692–703. IEEE.

Jiang, H., Weng, D., Zhang, Z., Bao, Y., Jia, Y., and Nie, M. (2018). Hikeyb: High-efficiency mixed reality system for text entry. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 132–137. IEEE.

Jiang, H., Weng, D., Zhang, Z., and Chen, F. (2019). Hifinger: One-handed text entry technique for virtual environments based on touches between fingers. *Sensors*, 19(14):3063.

Jones, A., Lang, M., Fyffe, G., Yu, X., Busch, J., McDowall, I., Bolas, M., and Debevec, P. (2009).

Achieving Eye Contact in a One-to-Many 3D Video Teleconferencing System. *ACM Transactions on Graphics (TOG)*, 28(3):1–8.

Kajita, H., Koizumi, N., and Naemura, T. (2016). Skyanchor: Optical design for anchoring mid-air images onto physical objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 415–423. ACM.

Kim, K., Billinghurst, M., Bruder, G., Duh, H. B.-L., and Welch, G. F. (2018). Revisiting Trends in Augmented Reality Research: a Review of the 2nd Decade of ISMAR (2008-2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2947–2962.

Kim, K., Bolton, J., Girouard, A., Cooperstock, J., and Vertegaal, R. (2012). TeleHuman: Effects of 3d Perspective on Gaze and Pose Estimation With a Life-Size Cylindrical Telepresence Pod. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2531–2540.

Kim, S., Lee, G., Huang, W., Kim, H., Woo, W., and Billinghurst, M. (2019). Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Kim, S., Son, J., Lee, G., Kim, H., and Lee, W. (2013). Tapboard: making a touch screen keyboard more touchable. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 553–562.

Kiyokawa, K., Billinghurst, M., Hayes, S. E., Gupta, A., Sannohe, Y., and Kato, H. (2002). Communication Behaviors of Co-located Users in Collaborative AR Interfaces. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pages 139–148. IEEE.

Knierim, P., Schwind, V., Feit, A. M., Nieuwenhuizen, F., and Henze, N. (2018). Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–9.

Kristensson, P.-O. and Zhai, S. (2004). Shark2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 43–52.

Kuester, F., Chen, M., Phair, M. E., and Mehring, C. (2005). Towards keyboard independent touch typing in vr. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 86–95.

Kumar, C., Hedeshy, R., MacKenzie, S., and Staab, S. (2020). Tagswipe: Touch assisted gaze swipe for text entry.

Kumar, M., Paepcke, A., and Winograd, T. (2007). Eyepoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 421–430.

Kunert, A., Kulik, A., Beck, S., and Froehlich, B. (2014). Photoportals: Shared References in Space and Time. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1388–1399. ACM.

Kunert, A., Weissker, T., Froehlich, B., and Kulik, A. (2019). Multi-Window 3D Interaction for Collaborative Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics.*

Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., and Betke, M. (2016). Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1952–1956.

Kurauchi, A. T. N. (2018). *EyeSwipe: text entry using gaze paths.* PhD thesis, Universidade de São Paulo.

Kurzhals, K., Göbel, F., Angerbauer, K., Sedlmair, M., and Raubal, M. (2020). A View on the Viewer:

Gaze-Adaptive Captions for Videos. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Kuster, C., Popa, T., Bazin, J.-C., Gotsman, C., and Gross, M. (2012). Gaze Correction for Home Video Conferencing. *ACM Transactions on Graphics (TOG)*, 31(6):1–6.

Lab, M. A. (2019). Sketch 2 Code. Transform Any Hands-Drawn Design Into a HTML Code With AI. https://sketch2code.azurewebsites.net.

Lakatos, D., Blackshaw, M., Olwal, A., Barryte, Z., Perlin, K., and Ishii, H. (2014). T (ether): Spatially-Aware Handhelds, Gestures and Proprioception for Multi-User 3D Modeling and Animation. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction*, pages 90–93. ACM.

Landgren, J. and Nulden, U. (2007). A Study of Emergency Response Work: Patterns of Mobile Phone Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1323–1332. ACM.

Layng, K., Gold, M., Ahlbrand, B., He, Z., and Perlin, K. (2020). The outpost. In *ACM SIGGRAPH 2020 Immersive Pavilion*, SIGGRAPH '20, New York, NY, USA. Association for Computing Machinery.

Le Goc, M., Kim, L. H., Parsaei, A., Fekete, J.-D., Dragicevic, P., and Follmer, S. (2016). Zooids: Building blocks for swarm user interfaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 97–109. ACM.

Lee, G., Kang, H., Lee, J., and Han, J. (2020). A User Study on View-Sharing Techniques for One-to-Many Mixed Reality Collaborations. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 343–352. IEEE.

Lee, G. A., Teo, T., Kim, S., and Billinghurst, M. (2018). A User Study on Mr Remote Collaboration Using Live 360 Video. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 153–164. IEEE.

Lee, L. H., Lam, K. Y., Li, T., Braud, T., Su, X., and Hui, P. (2019). Quadmetric optimized thumb-to-finger interaction for force assisted one-handed text entry on mobile headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–27.

Lee, M., Woo, W., et al. (2003). Arkb: 3d vision-based augmented reality keyboard. In *ICAT*.

Lee, S., Hong, S. H., and Jeon, J. W. (2002). Designing a universal keyboard using chording gloves. *ACM SIGCAPH computers and the physically handicapped*, (73-74):142–147.

Leithinger, D., Follmer, S., Olwal, A., and Ishii, H. (2014). Physical telepresence: shape capture and display for embodied, computer-mediated remote collaboration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 461–470. ACM.

Leithinger, D., Follmer, S., Olwal, A., and Ishii, H. (2015). Shape displays: Spatial interaction with dynamic physical form. *IEEE computer graphics and applications*, 35(5):5–11.

Li, F. C. Y., Guy, R. T., Yatani, K., and Truong, K. N. (2011). The 1line keyboard: a qwerty layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 461–470.

Li, J., Greenberg, S., Sharlin, E., and Jorge, J. (2014). Interactive Two-Sided Transparent Displays: Designing for Collaboration. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, pages 395–404. ACM.

Li, Z., Annett, M., Hinckley, K., Singh, K., and Wigdor, D. (2019). HoloDoc: Enabling Mixed Reality Workspaces That Harness Physical and Digital Content. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Liao, Y.-Y., Chou, L.-R., Horng, T.-J., Luo, Y.-Y., Young, K.-Y., and Su, S.-F. (2000). Force reflection and manipulation for a vr-based telerobotic system. *PROCEEDINGS-NATIONAL SCIENCE COUNCIL REPUBLIC OF CHINA PART A PHYSICAL SCIENCE AND ENGINEERING*, 24(5):382–389.

Lin, J.-W., Han, P.-H., Lee, J.-Y., Chen, Y.-S., Chang, T.-W., Chen, K.-W., and Hung, Y.-P. (2017). Visualizing the keyboard in virtual reality for enhancing immersive experience. In *ACM SIGGRAPH 2017 Posters*, pages 1–2.

Liu, Y., Zhang, C., Lee, C., Lee, B.-S., and Chen, A. Q. (2015). Gazetry: Swipe text typing using gaze. In *Proceedings of the annual meeting of the australian special interest group for computer human interaction*, pages 192–196.

Lombard, M., Ditton, T. B., and Weinstein, L. (2009). Measuring Presence: the Temple Presence Inventory. In *Proceedings of the 12th Annual International Workshop on Presence*, pages 1–15.

Lu, F., Yu, D., Liang, H.-N., Chen, W., Papangelis, K., and Ali, N. M. (2018). Evaluating Engagement Level and Analytical Support of Interactive Visualizations in Virtual Reality Environments. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 143–152. IEEE.

Lu, X., Yu, D., Liang, H.-N., Feng, X., and Xu, W. (2019). Depthtext: Leveraging head movements towards the depth dimension for hands-free text entry in mobile virtual reality systems. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1060–1061. IEEE.

Lu, Y., Yu, C., Yi, X., Shi, Y., and Zhao, S. (2017). Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–24.

Lukosch, S., Billinghurst, M., Alem, L., and Kiyokawa, K. (2015). Collaboration in Augmented Reality. *Computer Supported Cooperative Work (CSCW)*, 24(6):515–525.

Lutteroth, C., Penkar, M., and Weber, G. (2015). Gaze vs. mouse: A fast and accurate gaze-only click alternative. In *Proceedings of the 28th annual ACM symposium on user interface software & technology*, pages 385–394.

Ma, X., Yao, Z., Wang, Y., Pei, W., and Chen, H. (2018). Combining brain-computer interface and eye tracking for high-speed text entry in virtual reality. In *23rd International Conference on Intelligent User Interfaces*, pages 263–267.

MacKenzie, I. S., Kober, H., Smith, D., Jones, T., and Skepner, E. (2001). Letterwise: Prefix-based disambiguation for mobile text input. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 111–120.

MacKenzie, I. S. and Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 754–755.

MacKenzie, I. S. and Zhang, S. X. (1999). The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 25–31.

MacKenzie, I. S. and Zhang, X. (2008). Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 55–58.

Mahmood, T., Fulmer, W., Mungoli, N., Huang, J., and Lu, A. (2019). Improving Information Sharing and Collaborative Analysis for Remote GeoSpatial Visualization Using Mixed Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 236–247. IEEE.

Maimone, A., Yang, X., Dierk, N., State, A., Dou, M., and Fuchs, H. (2013). General-Purpose Telepresence With Head-Worn Optical See-Through Displays and Projector-Based Lighting. In *2013 IEEE Virtual Reality (VR)*, pages 23–26. IEEE.

Majaranta, P., Ahola, U.-K., and Špakov, O. (2009). Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 357–360.

Majaranta, P. and Räihä, K.-J. (2007). Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality*, pages 175–187.

Markussen, A., Jakobsen, M. R., and Hornbæk, K. (2014). Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1073–1082.

MASON, W. (2016). Nitero promises wireless desktop vr product by the "second half of 2016". http://uploadvr.com/nitero-wireless-vr-2016/.

McElree, B. and Carrasco, M. (1999). The temporal dynamics of visual search: evidence for parallel processing in feature and conjunction searches. *Journal of Experimental Psychology: Human Perception and Performance*, 25(6):1517.

McGill, M., Boland, D., Murray-Smith, R., and Brewster, S. (2015). A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2143–2152.

McNeely, W. A. (1993). Robotic graphics: A new approach to force feedback for virtual reality. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 336–341. IEEE.

Morris, M. R., Lombardo, J., and Wigdor, D. (2010). WeSearch: Supporting Collaborative Search and Sensemaking on a Tabletop Display. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 401–410. ACM.

Mott, M. E., Williams, S., Wobbrock, J. O., and Morris, M. R. (2017). Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2558–2570.

Neustaedter, C., Pang, C., Forghani, A., Oduor, E., Hillman, S., Judge, T. K., Massimi, M., and Greenberg, S. (2015). Sharing Domestic Life Through Long-Term Video Connections. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(1):1–29.

Nguyen, D. and Canny, J. (2005). MultiView: Spatially Faithful Group Video Conferencing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 799–808.

Nguyen, D. T. and Canny, J. (2007). Multiview: Improving Trust in Group Video Conferencing Through Spatial Faithfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1465–1474.

Norvig, P. (2013). English letter frequency counts: Mayzner revisited or etaoin srhldcu.

Oda, O., Elvezio, C., Sukan, M., Feiner, S., and Tversky, B. (2015). Virtual Replicas for Remote Assistance in Virtual and Augmented Reality. In *Proceedings of the 28th Annual ACM Symposium on UIST*, pages 405–415. ACM.

O'hara, K., Kjeldskov, J., and Paay, J. (2011). Blended Interaction Spaces for Distributed Team Collaboration. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18(1):3.

Ohkita, M., Obayashi, Y., and Jitsumori, M. (2014). Efficient visual search for multiple targets among categorical distractors: Effects of distractor–distractor similarity across trials. *Vision research*, 96:96–105.

Olofsson, J. (2017). Input and display of text for virtual reality head-mounted displays and hand-held positionally tracked controllers.

Olson, J. S., Olson, G. M., and Meader, D. K. (1995). What Mix of Video and Audio Is Useful for Small Groups Doing Remote Real-Time Design Work? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 362–368.

Orts-Escolano, S., Rhemann, C., Fanello, S., Chang, W., Kowdle, A., Degtyarev, Y., Kim, D., Davidson, P. L., Khamis, S., Dou, M., et al. (2016). Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM.

Otsuka, K. (2016). MMSpace: Kinetically-Augmented Telepresence for Small Group-to-Group Conversations. In *Virtual Reality (VR), 2016 IEEE*, pages 19–28. IEEE.

Otsuka, K. (2017). Behavioral Analysis of Kinetic Telepresence for Small Symmetric Group-to-Group Meetings. *IEEE Transactions on Multimedia*, 20(6):1432–1447.

Otsuka, K., Kumano, S., Ishii, R., Zbogar, M., and Yamato, J. (2013). Mm+ Space: Nx 4 Degree-of-Freedom Kinetic Display for Recreating Multiparty Conversation Spaces. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 389–396.

Otsuka, K., Kumano, S., Mikami, D., Matsuda, M., and Yamato, J. (2012). Reconstructing Multiparty Conversation Field by Augmenting Human Head Motions Via Dynamic Displays. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 2243–2248.

Otsuki, M., Sugihara, K., Kimura, A., Shibata, F., and Tamura, H. (2010). Mai painting brush: an interactive device that realizes the feeling of real painting. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 97–100. ACM.

Otte, A., Menzner, T., Gesslein, T., Gagel, P., Schneider, D., and Grubert, J. (2019). Towards utilizing touch-sensitive physical keyboards for text entry in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 1729–1732. IEEE.

Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O. (2013). Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2765–2774.

Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., and Hays, J. (2016). WebGazer: Scalable Webcam Eye Tracking Using User Interactions. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3839–3845. AAAI, AAAI.

Parizi, F. S., Whitmire, E., and Patel, S. (2019). Auraring: Precise electromagnetic finger tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(4):1–28.

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). GauGAN: Semantic Image Synthesis With Spatially Adaptive Normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1.

Paul, S. A. (2009). Understanding Together: Sensemaking in Collaborative Information Seeking.

Paul, S. A. and Morris, M. R. (2009). CoSense: Enhancing Sensemaking for Collaborative Web Search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1771–1780.

Pedersen, E. W. and Hornbæk, K. (2011a). Tangible bots: interaction with active tangibles in tabletop interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2975–2984. ACM.

Pedersen, E. W. and Hornbæk, K. (2011b). Tangible bots: Interaction with active tangibles in tabletop interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2975–2984, New York, NY, USA. ACM.

Pedrosa, D., Pimentel, M. D. G., Wright, A., and Truong, K. N. (2015). Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing (TACCESS)*, 6(1):1–37.

Pejsa, T., Kantor, J., Benko, H., Ofek, E., and Wilson, A. (2016). Room2room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 1716–1725.

Penkar, A. M., Lutteroth, C., and Weber, G. (2012). Designing for the eye: design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, pages 479–488.

Perlin, K. (1998). Quikwriting: continuous stylus-based text entry. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 215–216.

Perlin, K. (2016). Future reality: How emerging technologies will change language itself. *IEEE Computer Graphics and Applications*, 36(3):84–89.

Perlin, K., He, Z., and Rosenberg, K. (2018a). Chalktalk: a Visualization and Communication Language-As a Tool in the Domain of Computer Science Education. *ArXiv Preprint ArXiv:1809.07166*.

Perlin, K., He, Z., and Zhu, F. (2018b). Chalktalk vr/ar. *International SERIES on Information Systems and Management in Creative eMedia (CreMedia)*, (2017/2):30–31.

Perron, B. and Stearns, A. (2010). A review of a presentation technology: Prezi.

Pham, D.-M. and Stuerzlinger, W. (2019). Hawkey: Efficient and versatile text entry for virtual reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–11.

Piumsomboon, T., Dey, A., Ens, B., Lee, G., and Billinghurst, M. (2019). The effects of sharing awareness cues in collaborative mixed reality. *Frontiers in Robotics and AI*, 6:5.

Piumsomboon, T., Lee, G. A., Hart, J. D., Ens, B., Lindeman, R. W., Thomas, B. H., and Billinghurst, M. (2018). Mini-me: An adaptive avatar for mixed reality remote collaboration. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–13.

Popovici, I. and Vatavu, R.-D. (2019). Understanding Users' Preferences for Augmented Reality Television. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 269–278. IEEE.

Qin, R., Zhu, S., Lin, Y.-H., Ko, Y.-J., and Bi, X. (2018). Optimal-t9: An optimized t9-like keyboard for small touchscreen devices. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, pages 137–146.

Rajanna, V. and Hansen, J. P. (2018). Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*, pages 1–10.

Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. (1998). The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188. ACM.

Regenbrecht, H. and Langlotz, T. (2015). Mutual Gaze Support in Videoconferencing Reviewed. *CAIS*, 37:45.

Reilly, D., Tang, A., Wu, A., Mathiasen, N., Echenique, A., Massey, J., Rouzati, H., and Chamoli, S. (2011). *Toward a Framework for Prototyping Physical Interfaces in Multiplayer Gaming: TwinSpace Experiences*, pages 428–431. Springer Berlin Heidelberg, Berlin, Heidelberg.

Reilly, D. F., Rouzati, H., Wu, A., Hwang, J. Y., Brudvik, J., and Edwards, W. K. (2010). Twinspace: an infrastructure for cross-reality team spaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 119–128. ACM.

reklamistcomua (2019). World Builder (virtual Reality). https://www.youtube.com/watch?v=FheQe8rflWQ.

Richter, J., Thomas, B. H., Sugimoto, M., and Inami, M. (2007). Remote active tangible interactions. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 39–42. ACM.

Rick, J. (2010). Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 77–86.

Riedenklau, E., Hermann, T., and Ritter, H. (2012). An integrated multi-modal actuated tangible user interface for distributed collaborative planning. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 169–174. ACM.

Rosenfeld, D., Zawadzki, M., Sudol, J., and Perlin, K. (2004). Physical objects as bidirectional user interface elements. *IEEE Computer Graphics and Applications*, 24(1):44–49.

Ruan, S., Wobbrock, J. O., Liou, K., Ng, A., and Landay, J. A. (2018). Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–23.

Salimian, H., Brooks, S., and Reilly, D. (2019). MP Remix: Relaxed WYSIWIS Immersive Interfaces for Mixed Presence Collaboration With 3D Content. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–22.

Sarcar, S., Panwar, P., and Chakraborty, T. (2013). Eyek: an efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th asia pacific conference on computer human interaction*, pages 215–220.

Satriadi, K. A., Ens, B., Cordeil, M., Jenny, B., Czauderna, T., and Willett, W. (2019). Augmented Reality Map Navigation With Freehand Gestures. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 593–603. IEEE.

Schrepp, M., Hinderks, A., and Thomaschewski, J. (2017). Construction of a Benchmark for the User Experience Questionnaire (UEQ). *IJIMAI*, 4(4):40–44.

Sellen, A. J. (1995). Remote Conversations: the Effects of Mediating Talk With Technology. *Human-Computer Interaction*, 10(4):401–444.

Shi, W., Yu, C., Yi, X., Li, Z., and Shi, Y. (2018). Toast: Ten-finger eyes-free typing on touchable surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):1–23.

Sindhwani, S., Lutteroth, C., and Weber, G. (2019). Retype: Quick text editing with keyboard and gaze. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Sirkin, D., Venolia, G., Tang, J., Robertson, G., Kim, T., Inkpen, K., Sedlins, M., Lee, B., and Sinclair, M. (2011). Motion and Attention in a Kinetic Videoconferencing Proxy. In *IFIP Conference on Human-Computer Interaction*, pages 162–180. Springer, Springer.

Smith, B. A., Bi, X., and Zhai, S. (2015). Optimizing touchscreen keyboards for gesture typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3365–3374.

Software, T. C. (2019). Miro Collaboration Without Constraints. https://Miro.com/.

Soukoreff, R. W. and MacKenzie, I. S. (2003). Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120.

Speer, R., Chin, J., Lin, A., Jewett, S., and Nathan, L. (2018). Luminosoinsight/wordfreq: v2.2.

Speicher, M., Feit, A. M., Ziegler, P., and Krüger, A. (2018). Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Sra, M. (2016). Asymmetric design approach and collision avoidance techniques for room-scale multiplayer virtual reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 29–32. ACM.

Sra, M., Jain, D., Caetano, A. P., Calvo, A., Hilton, E., and Schmandt, C. (2016). Resolving spatial variation and allowing spectator participation in multiplayer vr. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 221–222. ACM.

Sra, M., Mottelson, A., and Maes, P. (2018a). Your Place and Mine: Designing a Shared VR Experience for Remotely Located Users. In *Proceedings of the 2018 Designing Interactive Systems Conference*, DIS '18, pages 85–97, New York, NY, USA. ACM.

Sra, M., Mottelson, A., and Maes, P. (2018b). Your Place and Mine: Designing a Shared VR Experience for Remotely Located Users. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pages 85–97.

Sra, M. and Schmandt, C. (2015). Metaspace: Full-body tracking for immersive multiperson virtual reality. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 47–48. ACM.

Sridhar, S., Feit, A. M., Theobalt, C., and Oulasvirta, A. (2015). Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3643–3652.

Steptoe, W., Wolff, R., Murgia, A., Guimaraes, E., Rae, J., Sharkey, P., Roberts, D., and Steed, A. (2008). Eye-Tracking for Avatar Eye-Gaze and Interactional Analysis in Immersive Collaborative Virtual Environments. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, pages 197–200.

Stone, R. (2001). Haptic feedback: A brief history from telepresence to virtual reality. *Haptic Human-Computer Interaction*, pages 1–16.

Sugihara, K., Otsuki, M., Kimura, A., Shibata, F., and Tamura, H. (2011). Mai painting brush++: Augmenting the feeling of painting with new visual and tactile feedback mechanisms. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, pages 13–14. ACM.

Sun, Q., Wei, L.-Y., and Kaufman, A. (2016). Mapping virtual and physical reality. *ACM Transactions on Graphics (TOG)*, 35(4):64.

Surname, F. (2018). Text input in virtual reality using a tracked drawing tablet.

Tam, T., Cafazzo, J. A., Seto, E., Salenieks, M. E., and Rossos, P. G. (2007). Perception of Eye Contact in Video Teleconsultation. *Journal of Telemedicine and Telecare*, 13(1):35–39.

Tan, K.-H., Gelb, D., Samadani, R., Robinson, I., Culbertson, B., and Apostolopoulos, J. (2010). Gaze Awareness and Interaction Support in Presentations. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 643–646, New York, NY, USA. ACM.

Tang, A., Pahud, M., Inkpen, K., Benko, H., Tang, J. C., and Buxton, B. (2010). Three's Company: Understanding Communication Channels in Three-Way Distributed Collaboration. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, pages 271–280. ACM.

Tang, J. C. and Minneman, S. (1991). VideoWhiteboard: Video Shadows to Support Remote Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 315–322. ACM.

Tang, J. C. and Minneman, S. L. (1990). VideoDraw: a Video Interface for Collaborative Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 313–320. ACM.

Teo, T., Lawrence, L., Lee, G. A., Billinghurst, M., and Adcock, M. (2019). Mixed reality remote collaboration combining 360 video and 3d reconstruction. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–14.

Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., Pandey, R., Fanello, S., Wetzstein, G., Zhu, J.-Y., Theobalt, C., Agrawala, M., Shechtman, E., Goldman, D. B., and Zollhöfer, M. (2020). State of the Art on Neural Rendering. *Eurographics*, 39(2).

Thanyadit, S., Punpongsanon, P., and Pong, T.-C. (2018). Efficient Information Sharing Techniques Between Workers of Heterogeneous Tasks in 3d Cve. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–19.

Thanyadit, S., Punpongsanon, P., and Pong, T.-C. (2019). ObserVAR: Visualization System for Observing Virtual Reality Users Using Augmented Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 258–268. IEEE.

Thomas, B. H. and Piekarski, W. (2002). Glove based user interaction techniques for augmented reality in an outdoor environment. *Virtual Reality*, 6(3):167–180.

Tversky, B., Suwa, M., Agrawala, M., Hanrahan, P., Phan, D., Klingner, J., Daniel, M., Lee, P., and Haymaker, J. (2003). Human Behavior in Design: Individuals, Teams, Tools. *Sketches for Design and Design of Sketches*, pages 79–86.

Vertanen, K., Fletcher, C., Gaines, D., Gould, J., and Kristensson, P. O. (2018). The impact of word, multiple word, and sentence input on virtual keyboard decoding performance. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Vertanen, K., Memmi, H., Emge, J., Reyal, S., and Kristensson, P. O. (2015). Velocitap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 659–668.

Vertegaal, R. (1999). The gaze groupware system: Mediating joint attention in multiparty communication and collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 294–301, New York, NY, USA. ACM.

Vertegaal, R., Weevers, I., Sohn, C., and Cheung, C. (2003). GAZE-2: Conveying Eye Contact in Group Video Conferencing Using Eye-Controlled Camera Direction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 521–528.

Vinge, V. (2006). Rainbows end: A novel with one foot set in the future.

Voelker, S., Hueber, S., Holz, C., Remy, C., and Marquardt, N. (2020). GazeConduits: Calibration-Free Cross-Device Collaboration Through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Vogt, K., Bradel, L., Andrews, C., North, C., Endert, A., and Hutchings, D. (2011). Co-located Collaborative Sensemaking on a Large High-Resolution Display With Multiple Input Devices. In *IFIP Conference on Human-Computer Interaction*, pages 589–604. Springer, Springer.

VOID, T. (2016). The void: The vision of infinite dimensions. https://thevoid.com/.

Volmer, B., Baumeister, J., Von Itzstein, S., Bornkessel-Schlesewsky, I., Schlesewsky, M., Billinghurst, M., and Thomas, B. H. (2018). A Comparison of Predictive Spatial Augmented Reality Cues for Procedural Tasks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2846–2856.

VRcade, I. (2015). The vrcade: A virtual reality platform for truly immersive gaming. http://vrcade.com/.

Walker, J., Li, B., Vertanen, K., and Kuhl, S. (2017). Efficient typing on a visually occluded physical keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5457–5461.

Wallace, J. R., Scott, S. D., Stutz, T., Enns, T., and Inkpen, K. (2009). Investigating Teamwork and Taskwork in Single-And Multi-Display Groupware Systems. *Personal and Ubiquitous Computing*, 13(8):569–581.

Wang, J., Zhao, K., Zhang, X., and Peng, C. (2014). Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 14–27.

Ward, D. J., Blackwell, A. F., and MacKay, D. J. (2000). Dasher—a Data Entry Interface Using Continuous Gestures and Language Models. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 129–137.

Weichel, C., Lau, M., Kim, D., Villar, N., and Gellersen, H. W. (2014). MixFab: a Mixed-Reality Environment for Personal Fabrication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3855–3864.

Weissker, T., Kulik, A., and Froehlich, B. (2019). Multi-Ray Jumping: Comprehensible Group Navigation for Collocated Users in Immersive Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 136–144. IEEE.

Whitmire, E., Jain, M., Jain, D., Nelson, G., Karkar, R., Patel, S., and Goel, M. (2017). Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–21.

Wong, P. C., Zhu, K., and Fu, H. (2018). Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–10.

Wu, P.-C., Wang, R., Kin, K., Twigg, C., Han, S., Yang, M.-H., and Chien, S.-Y. (2017). DodecaPen: Accurate 6DoF Tracking of a Passive Stylus. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 365–374. ACM.

Xu, L.-Q., Loffler, A., Sheppard, P., and Machin, D. (1999). True-View Videoconferencing System Through 3-D Impression of Telepresence. *BT Technology Journal*, 17(1):59–68.

Xu, W., Liang, H.-N., He, A., and Wang, Z. (2019a). Pointing and selection methods for text entry in augmented reality head mounted displays. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 279–288. IEEE.

Xu, W., Liang, H.-N., Zhao, Y., Zhang, T., Yu, D., and Monteiro, D. (2019b). Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE transactions on visualization and computer graphics*, 25(5):1991–2001.

Xu, Z., Chen, W., Zhao, D., Luo, J., Wu, T.-Y., Gong, J., Yin, S., Zhai, J., and Yang, X.-D. (2020). Bitiptext: Bimanual eyes-free text entry on a fingertip keyboard. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Yanagihara, N., Shizuki, B., and Takahashi, S. (2019). Text entry method for immersive virtual environments using curved keyboard. In *25th ACM Symposium on Virtual Reality Software and Technology*, pages 1–2.

Yi, J. S., Kang, Y.-a., Stasko, J. T., and Jacko, J. A. (2008). Understanding and Characterizing Insights: How Do People Gain Insights Using Information Visualization? In *Proceedings of the 2008 Workshop on BEyond Time and Errors: Novel EvaLuation Methods for Information Visualization*, page 4. ACM.

Yi, X., Wang, C., Bi, X., and Shi, Y. (2020). Palmboard: Leveraging implicit touch pressure in statistical decoding for indirect text entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Yi, X., Yu, C., Zhang, M., Gao, S., Sun, K., and Shi, Y. (2015). Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 539–548.

Yin, Y., Li, Q., Xie, L., Yi, S., Novak, E., and Lu, S. (2016). Camk: A camera-based keyboard for small mobile devices. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE.

Yokokohji, Y., Hollis, R. L., and Kanade, T. (1996). What you can see is what you can feel-development of a visual/haptic interface to virtual environment. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*, pages 46–53. IEEE.

Young, J., Langlotz, T., Cook, M., Mills, S., and Regenbrecht, H. (2019). Immersive Telepresence and Remote Collaboration Using Mobile and Wearable Devices. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1908–1918.

Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., and Shi, Y. (2017). Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4479–4488.

Yu, C., Sun, K., Zhong, M., Li, X., Zhao, P., and Shi, Y. (2016). One-dimensional handwriting: Inputting letters and words on smart glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 71–82.

Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W., and Liang, H.-N. (2018). Pizzatext: text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2927–2935.

Zhai, S., Hunter, M., and Smith, B. A. (2000). The metropolis keyboard-an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 119–128.

Zhai, S., Hunter, M., and Smith, B. A. (2002). Performance optimization of virtual keyboards. *Human–Computer Interaction*, 17(2-3):229–269.

Zhai, S. and Kristensson, P.-O. (2003). Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 97–104, New York, NY, USA. ACM.

Zhang, M. R., Wen, H., and Wobbrock, J. O. (2019). Type, then correct: Intelligent text correction techniques for mobile text entry using neural networks. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pages 843–855.

Zhu, J., Yang, R., and Xiang, X. (2011). Eye Contact in Video Conference Via Fusion of Time-of-Flight Depth Sensor and Stereo. *3D Research*, 2(3):5.

Zhu, S., Luo, T., Bi, X., and Zhai, S. (2018). Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.