Doctoral Dissertation
Academic Year 2018

Convex Interactions: Towards Efficient Human Motion in Peripersonal Space Using Virtual Reality

Keio University
Graduate School of Media Design

Yun Suen Pai

A Doctoral Dissertation
submitted to Keio University Graduate School of Media Design
in partial fulfillment of the requirements for the degree of
Ph.D. in Media Design

Yun Suen Pai

Thesis Advisory Committee:

Associate Professor Kai Kunze	(Supervisor)
Associate Professor Kouta Minamizawa	(Member)
Assistant Professor Liwei Chan	(Member)
Professor Keiko Okawa	(Member)

Thesis Review Committee:

Associate Professor Kouta Minamizawa	(Member)
Assistant Professor Liwei Chan	(Member)
Professor Keiko Okawa	(Member)
Professor Masahiko Inami	(Member)

Abstract of Doctoral Dissertation of Academic Year 2018

Convex Interactions: Towards Efficient Human Motion in Peripersonal Space Using Virtual Reality

Category: Science / Engineering

Summary

A human's everyday interactions with the environment and each other are the product of many years of evolution, as we understand the spaces around us and decide how best to perform the appropriate motions and gestures that are befitting for every scenario. These interactions that we perform are out of intuition; the way our legs move to propel ourselves forward, the way our fingers close when grasping, and so on. These movements have become so natural that, we rarely question the possibility of the next form of interaction that could possibly be more efficient that what we deem natural. However, with the advancements in human-computer interaction (HCI), technology can be used to help us design and develop interaction mechanics that could possibly shape how we perform all our interactions for the better as man marry machine.

To understand this, I look into the concept of space and bodily mapping to define new motions and gestures for everyday interactions. Convex Interactions, defined in this thesis, are interaction mechanics that utilize our proxemic and peripersonal space sense to shortcut interactions, both spatially and through exploration of bodily mapping, to create a more efficient and intuitively superior form of interaction than what we are used to. I used virtual reality (VR) and physiological signal input as a tool to design Convex Interactions. I explore 1) the possibility of shortcutting an interaction that normally involves multiple gesture into only one by reducing the space of interaction to within the intimate space of the user, 2) how different mapped input of a motion can still correlate with what we perceive as natural yet more space efficient, 3) how further reducing the space of interaction to within the peripersonal space while directly changing the output still can feel intuitively superior, and 4) other forms of human interaction that can benefit from this increased efficiency and shortcut in interaction. Finally, I discuss possible scenarios where Convex Interactions can be used beyond VR and as the next evolution of human input and interaction with the use of human-assisted machines through extending body schemas and micro movements.

Keywords:

Convex Interactions, Virtual Reality, Human Motion, Physiological Signals, Peripersonal Space

Keio University Graduate School of Media Design,

Yun Suen Pai

Contents

List of Figures

List of Tables

Acknowledgements

I am eternally grateful to my supervisor, Kai Kunze, for always being supportive for all the random things I have done throughout my Ph.D. We literally shared the same work space these past few years, yet you have been so patient with my shenanigans and always reply me with constant motivation and your immense knowledge in the field.

I am also grateful to my review committees, namely Kouta Minamizawa, Liwei Chan, Keiko Okawa, and Masahiko Inami for always striving to make me produce the best thesis possible. This process is never a smooth one, so thank you for the patience and believe you guys had in me, even when I doubted myself.

A special thanks to those who have collaborated with me these past three years. Benjamin Outram, Benjamin Tag, Liwei Chan, Tilman, and Takuro especially, you guys are the best! I would also like to apologize for the times where my stubbornness got in the way of producing something greater, yet you guys still continue to work with me through sleepless nights.

My eternal gratitude to my family back home who, even though can't be with me physically, were always supportive of my work, both morally and emotionally. To my late grandmother, whom I regret not being able to visit at your last moments because of my studies; may you be in a better place up there as you watch over me. Also to my friends, both back home and here, who helped keep my sanity in check simply from casual small talks, you guys have no idea how important that is in this perilous journey.

Finally, I would never have succeeded without the continuous support from Ivy and Mio who were always there for me through thick and thin. The times we spend together has defined me throughout my Ph.D. journey, and I hope our little group never ends!

Chapter 1

Introduction

In this section, I briefly explain the history of human interaction and how we transition from that to HCI where each of our input is translated from physical space to digital space. Then, I explore the definition of efficient interactions and how human evolution shows us that our motions and gestures prioritize efficiency. I end the introduction with discussing the primary presented research question in this thesis.

1.1. Spaces of Interaction

When humans are first born to this world, interactions immediately become something that we need to engage in. As an infant, we first learn to breathe and cry, followed by more complex interactions like grabbing things with our hands. In time, we gain a sense of space around us, as we know if something is close or far to us, or if an interaction with an item or person will create different results. Being in a warm water feels relaxing, whereas water that is too hot causes us to cry, as this information becomes embedded in our minds to not reach out and touch a possibly hot body of water.

The concept of space pushes the boundaries of many fields, from HCI to psychology, because it quite literally engulfs our daily activities and our daily lives. A single objective definition of space varies according to the field of research. We use the concepts of spaces to think about interactive experiences [12]. An interactive experience is dependant on the technology being used as well as the digital content. With regards to digital content, which range from photos, to videos and virtual worlds, e need to take into account the both the asset of the content as well as the function to allow people to access and process them [12]. Outside of the digital space however, our interaction space is simply what we do everyday with the people and objects around us. In the physical space, interactions can be conceptualized into people, activity, context and technologies (PACT). Furthermore, space itself can be further divided into physical space, digital space, information space, conceptual space, social space, navigating space, and blended space. In the interest of this thesis, we focus more on physical and digital space, and the connection between them.

1.1.1 Physical and Digital Space

The study on physical space and the interaction with them can be further explored within the field of social psychology and neuropsychology. When looking at human space, the field of social psychology defined the term proxemics. The definition of proxemics will be further explained in Chapter 2. Essentially, proxemics is about the study of human space which can be divided into intimate, personal, social, and public space with increasing area [50]. The amount of space taken for each of these tiers are objectively defined. Basically, the closer we are to another person's intimate space, the higher the possibility of being intrusive to that person. Likewise, the further we move our body parts around, the higher the chance of it entering someone else's intimate space and causing intrusion. Another definition can be found in the field of neuropsychology with respect to peripersonal space. In this field, the space around us can be divided into two types; peripersonal space and extrapersonal space. More detailed explanation will be given in Chapter 2 as well, though essentially, unlike proxemics, peripersonal space is not objectively defined. It is closer than intimate space in proxemics, and constantly changes depending on the actions and senses of the person. It is closely related to the definition of spatial awareness as opposed to the sense of personal space that leads to intrusion.

For the digital space, due to the wide variation of technological devices we interact with, this topic needs to be explored step by step. Digital space refers to the world of VR, excel sheets, music, Facebook, and so on. A good way to differentiate digital space with physical space is, unlike physical space, it is the space of bits rather than atoms [12]. Because of the vastness of digital devices, it is a challenge to summarize how each of them interact. For example, a space for screen-based devices like a computer, depends on the amount of pixels that are present in the display and how the user interface utilizes them. Since it is software based, many possible tweaks can be used to utilize this space. Smaller physical space like a smartphone device may use multiple home screens to further extend the available space of interaction. However, they do become more ubiquitous everyday, where our smartphones can communicate with our television, our computer, our wrist wactches and so on. Regarding the connection between physical and digital space, this can be defined as a cyber-physical system. Such a system are physical spaces that use interactions in a digital space to reflect its functionality. A good example of such a medium would be VR, where we perform physical based interactions to interact with digital content.

1.1.2 Virtual Reality Space

When Jaron Lanier [77] used the term virtual reality, he referred to it as a system that allows users to experience a shared virtual environment with interactable models that fully engulfs the users FOV. Essentially, VR is a form of computer technology that immerses the user in a virtual environment or a computer-generated world with the use of realistic images and sound. This technology allows the user to experience a form of telepresence that places him or her in the shoes of another, or augment the visual environment with virtual content. VR nowadays usually comprises of a head mounted display (HMD), aided by several peripherals that enable other forms of tracking. The HMD itself provides head tracking for the user so that he or she can physically look around as in real life, as opposed to devices that merely recreates the screen digitally as an enhanced or larger version (often called cinema mode). This accurate tracking with low latency in essence, defined the current generation of VR devices.

This also provides total immersion in terms of sight for the user, since it almost completely covers the users FOV. Prior to VR, several desktop applications utilize a first-person view camera, which basically displays the screen according to what the virtual avatar would see. However, that was the main issue; it was according to the avatars vision as opposed to the actual user, who still watches the monitor or screen that displays the content. Therefore, HMDs basically strap the display onto the users eyes and uses various sensors to determine the orientation of the device. Older HMDs had high latency and a relatively narrow FOV, which breaks the immersion and can actually cause motion sickness. VR devices today are a vast improvement over that, while also incorporating additional sensing modalities such as microphone, eye tracking, and room tracking. Peripherals then complements the existing HMD devices, allowing users to import their hand or even full body presence into the virtual world with high tracking accuracy.

Generally, the virtual content being displayed places the user in environments that is only limited by the developers imagination. Depending on the use case, VR has been utilized to recreate a war zone, surgical room, flight cockpit, construction site, outer space and many more. Additionally, these technologies allow for modifications that are not possible for other forms of medium, such as changing the sense of embodiment, sense of scale, sense of presence, and so on. Because of this, the potential applications are nearly limitless. The human mind is a fragile thing, and with these technologies being able to trick it enough towards total immersion, it can even be used for applications like training or even therapy. The possibilities are near endless, limited only by the developer's imaginations.

Despite the existence of VR for the past decades, they never kicked off as a consumer product until 2012 when Oculus launched their Kickstarter campaign to fund the development of the VR HMD dubbed the Oculus Rift. Their first version, which was the Development Kit 1 (DK1) finally shipped to backers during mid-2013, and it was the first affordable and reliable VR experience for consumers. Even though it was only aimed towards enthusiasts and developers, it did not stop most people from getting their hands on a unit due to its affordable price. The DK1 may not be the first ever VR headset that was commercially available, but it achieved that at a stellar performance and affordable price. From that point on, VR was starting to be adopted by many as this unique experience unlike any other.

Noticing that the DK1s adoption was high due to its relatively affordable price (though its availability was not the best for certain countries), Google decided to bank on the idea of an extremely affordable VR experience that literally anyone could experience. They created the Google Cardboard, which was a HMD made entirely from cardboard, yet still delivers an acceptable, albeit slightly downgraded, VR experience, on 2014. It leverages the sensors in our smartphones as a display system for the cardboard, ensuring that almost anyone with a smartphone will be able to just purchase a cardboard and literally experience VR immediately. Even though the latency and performance downgrades are noticeable compared to standalone VR HMDs like the DK1, it was not created for the purpose of being better, but rather to increase VR usage and acceptance among the public. 2014 was also the year of which Oculus released their updated Rift DK2, with improvements such as a higher resolution screen, higher refresh rate, and positional tracking. It became the definitive VR experience for enthusiasts and developers, and no other HMDs were close in terms of performance and price.

However, looking at the success of Google Cardboard, Oculus realized the impact of targeting the mass audience with an extremely accessible VR experience. Therefore on 2015, they collaborated with Samsung to release the Gear VR, which function similarly like Google Cardboard, with a few additional features for Samsungs Galaxy S smartphone series. This too, proved to be a hit, since Samsung controls the market share for Android phones as of this moment. It costs more than the Cardboard, but was cheaper and more accessible than the DK2.

The world of VR begun to fire up once again on 2016, which many claim to be the first year VR emerges as a development platform. Prior to this, Oculus has been the only company that created the VR HMD that all VR enthusiasts would buy. However, HTC, in collaboration with Valve, released the HTC Vive which took VR to a new level, to directly compete with Oculuss latest offering, the CV1. Furthermore, Sony joined the VR ecosystem with the release of the Playstation VR, turning 2016 into arguable the most exciting year for anyone in the VR field. Starting with the launch of the CV1, the improved HMD by Oculus offered an overall better design, higher resolution, and better tracking. Oculus also showed consumers the Oculus Touch, which were a pair of controllers that allows hand presence in VR. Hand presence is actually the evolution of VR in 2016 that can further provide a higher immersion level that will render traditional gamepads obsolete. However, the Oculus Touch was not launched until late 2016. On the other hand, the HTC Vive launched with critical acclaim due to its room tracking features. Room tracking refers to its ability to track the user in a room-scale, complete with hand tracking via the Vive controllers. It revolutionalized how we use VR today, and blew the minds of anyone who experience VR for the first time. In many ways, the HTC Vive is largely comparable to the CV1, however, the inclusion of room tracking and hand presence made it overall more successful than the CV1 at launch, despite it being more expensive. However, the biggest caveat suffered by both the Vive and CV1 is the cost. Purchasing the CV1 costs around $600, while the HTC Vive costs $800. Furthermore, they both require a powerful desktop computer to run the system, which is an additional $1500 to $2000. This highly limits the adoption of VR and only caters to those who wish to be in front of the rest in experiencing this cutting edge technology. Thus, the emergence of Playstation VR was meant to fix this underlying issue. It costs much less than both aforementioned devices and it runs directly on any Playstation 4 which costs much less that a VR-ready desktop. It was also developed with a plug-and-play feature in mind, since it was catered solely to the gaming audience. The key factor here is the target towards mainstream audience as opposed to enthusiasts, which also means it can be more widely used. Mainstream will be an ongoing term throughout this research as a VR system that appeals to the highest amount of audience stands the highest chance to be used by virtually anyone, anytime, anywhere. Such a future is debatable to be desirable or not, but remains a fact that it could very well be the future of media consumption on the go.

The space around us is an important consideration for an interaction mechanic. This is evident when booting up SteamVR for the HTC Vive in the case of VR; it requires the user to first clear the room of any nearby furniture or objects before engaging with the system. This is due to the nature of these systems that envelope our peripheral visions and sense of space, making it difficult to discern our proximity with nearby physical objects. Even though most of the current AR/VR systems available today integrates some kind of tool to map the environment and create a virtual barrier for the player to see, most often than not, players will still exceed the confines of the barrier in the heat of the immersion. It is relatively easy for us to move these static objects away from the play area, but not so for dynamic objects such as the presence of other players.

However, for the space in the virtual world, it actually somehow provides us with a real sense of space when we interact or move through them. With the introduction of additional peripherals, whether it be first or third party, our involvement with the virtual world gradually increases. For example, the use of a threadmill for walking in digital space, without actually using much physical space, gives us a strong sense of presence. In fact, navigation is one of the greatest examples to use for spatial use of physical and virtual space. An initial model was developed by Steve Benford and Lenart Fahln [10] for navigation where they found that when users interact with each other or with virtual objects, a medium is usually improved. This medium is a form of aura around virtual objects and virtual avatars that gives each user the sense of presence. This aura, similar to physical space, is like the peripersonal space previously mentioned in neuropsychology. Because of this, VR is a great platform to emulate and simulate interactions in the real world. Still, I will next explore interactions that we deem as natural, and establish VR's contribution in making our interactions more efficient.

1.2. Beyond Natural Interactions

Bill Buxton, a principal researcher at Microsoft's Natural User Interface group, said that "rather than technological or inate in the human, what is natural has to do with specific skills (motor-sensory, cognitive, social, cultural and emotional) that the user has acquired through a life-time of living in the everyday world." The definition of "natural interaction" varies according to context, but it typically means that an interaction is close to that of the physical world counterpart [112]. It could also mean that an interaction mechanic has a high learning rate to quickly transition from novice to mentor, often deemed as a Natural User Interface (NUI). In the end, an interaction is deemed natural because it is defined based on experience: in real life, humans communicate not just vocally, but through gestures, expressions, and movements, as we discover the world through manipulation of the physical matter around us [157]. Other related works also link gestures towards natural interactions [24, 163]. Gestures, interestingly, exists in many forms with some of them subtler than others. One of them exists in the form of physiological signals, which can be used as a form of natural interaction [89]. However, in the end of the day, based on these related work, we can see that the definition of natural is very important in HCI, yet remains extremely subjective by nature. Based on the previously mentioned quote by Bill Buxton, what is natural simply depends on the user's personal experience as well as the presented scenario.

If an interaction is deemed natural to a user, is it also intuitive? Intuitive, by definition, refers to perceiving directly by intuition without rational though, as a person or the mind [2]. Because of the lower cognitive load when performing an interaction, an intuitive interface is often synonymous with familiarity. However, therein lies the issue with designing a new interaction or user interface; familiarity destroys novelty. What most users find to be intuitive, is, like natural, depends on what they know from their previous experience. Because of this, it becomes difficult to develop something that is novel yet remains intuitive, because novelty is about something new where others have not been exposed to it yet. Therefore, to stick to a design that is natural and/or intuitive is not wise, as this thesis looks into the realm of interaction beyond that.

Looking back at the concept of navigation as a form in interaction for humans, I will again discuss about how we interact with things the moment we are born. For navigation, we started with crawling. Based on how we define what is natural then, when we were an infant, crawling can be considered as the most natural way we interact with our space; we crawl to where we want to go to. Eventually though, we moved towards walking with our legs. By that time, we look at walking with legs only as the most natural way of navigating, not returning to crawling anymore. However, we intuitively and instinctively started with crawling. Does this mean that crawling can be considered natural? When we were infants, maybe. However, we evolved to walking with our legs not just from human evolution, but simply the fact that walking with legs achieves the same thing as crawling, but with less limbs required, as well as less horizontal space. In other words, we find walking with legs to be more efficient than crawling.

If we look at digital space again and try to establish its connection with navigational input, we can see that each of them are tied to its specific input device. Navigation in this regard, refers to actual movement or walking; the walkign motion of a real human, or a virtual avatar. For a computer, where our input device is a keyboard and mouse, when controlling a virtual character to move, we use the keyboard buttons. In a gaming console, where the input device is primarily a gamepad, we tilt the analogue stick towards the direction we wish our avatar to go. For VR, where current input are motion-tracked controllers, it is actually safe to say that there is not predefined locomotion mechanic yet. However, given the input device as well as worries regarding motion sickness, most VR applications have resorted to a teleportation mechanic where we point with a controller and press a button to teleport to the location we point to. Can we say that each of these methods are intuitive or natural, since none of them are actually similar to how we walk? Perhaps not, however, we can say that these methods are efficient at achieving the task; so much so that we have growned accustomed to these methods over time. These methods are efficient; they require minimal motion, less energy, and less motion space to achieve locomotion compared to physical walking.

This also presents another interesting research question; if buttons have proven its efficiency, then can it be used to substitute our everyday interactions? Perhaps, but again, I will look into navigation or locomotion as an example of interaction here. Studies in VR has shown that this causes motion sickness, leading to many methods VR developers are forced to use to reduce this [42]. This does not disregard buttons because it is obviously useful for specific task like turning on a switch, but not so for more general human interactions where motion is preferable. From each of the previously mentioned digital spaces, VR is clearly the closest there is to physical space interaction. Of course we do not teleport to navigate, but since its motion controllers and HMD captures our every movement, it is a suitable platform to envision the next step of efficient interaction. Therefore, I establish again that VR is a suitable platform to let us understand the balance between what we find effective and what we find to be natural or intuitive.

1.2.1 Physiological Signal Input

This section will discuss the method of sensing the aforementioned motions and gestures. When we previously stress on motion controllers for VR, these are controllers that are able to sense our motions and gestures. The signals from the human body that correlates with out movements are the physiological signals. Physiological data refers to the electrical signals that human generate when they perform everyday activities like speaking, blinking, moving, and so on. These data can be read using specific sensors and displayed for us to understand and quantify our physiological state. Utilizing these sensing methods also involves the rethinking of new ways of how they work in a virtual environment. This is where one of the main benefits of physiological signals become apparent. These signals allow hands-free control for VR [122], opening many possibilities for VR interactions in a subtle, non-intrusive manner. This also means it will not hinder conventional motion-tracked controllers, making both input methods viable depending on application.

The next benefit of physiological sensing is that it can be reflected in a virtual environment both explicitly and implicitly. As an input modality, physiological sensing provides the user with an explicit sense of control unlike any other; imagine pulling a virtual trigger of a gun using our own muscle contraction, which in turn provides the illusion of the guns recoil. However, it is the implicit factor that makes these sensing methodologies unique. [133]. Physiological sensing can either be directly manipulated by us, or is something that naturally reflects our current state, such as an increase in heart rate. This provides us with an additional layer of information or feedback from the user that more accurately reflects the thought, as opposed to explicit feedback like vocal expression which is arguably inaccurate at times. By reflecting these implicit feedback in a virtual scene, this opens up a new frontier of AR/VR development that delves into human embodiment and augmentation. Implicit feedback or output can generate content that is not direct, but rather implied by the user through their own physiological signal. Often, such a feedback can also provide information that the user's themselves do not even know. These signals transcend verbal or any form of direct information delivery, and opens up many more possibilities for AR/VR design.

1.2.2 Efficiency or Intuitiveness?

Efficiency has been mentioned several times in this chapter, and now I shall delve into its definition. Efficiency can be described as the ratio of output to the input of a given system. In interaction design, high efficiency is when the user reaches a goal with as few resources as possible [57]. What then, are the resources of an interaction? When we perform an input to a system, whether it is an AR/VR environment or a wearable device, the resource that we use is the time to achieve the goal as well as the space and movement needed to achieve it. With regards to time, microinteractions have been defined as an interaction that can be achieved in no more than 4 seconds [6]. With regards to space and movement however, it has not been clearly defined. If an interaction achieves its goal but with a lesser amount of space, it can be said that the interaction is more efficient.

Therefore, we have looked into the definition of natural, intuitive, and efficient. An interaction is natural when it is close to what we are used to in a given scenario. An interaction is intuitive if we are very familiar with it. Both these methods though, may lack novelty. Finally, an interaction is efficient when it requires less resources. From all the previously mentioned examples, we can see that how we interact or even use a device can depend on how efficient it is (button for computer, gamepad for gaming consoles, etc.), even if it may not be natural. On the other hand, if something is completely unnatural, we as humans may also naturally reject it (feeling of motion sickness for button or gamepad-based locomotion in VR), making it also clear that to a certain degree, interactions should still be natural. Therefore, we need to find balance; an interaction that can possible be more efficient than what we already deem as natural, yet is still able to preserve to a certain degree, some amount of naturalness where it can eventually be intuitive to us. If such a scenario can happen, is there a possibility for this interaction to even overwrite what we are used to the point that we can claim it as the next level of natural interaction?

1.3. Correlated Mapping

In this section, we discuss about the possibility of an interaction that borrows an element of naturalness, yet still maintains some novelty which may, or may not, present additional benefits in that given scenario.

To use physiological sensing methods to create natural interactions, we generally aim to map our inputs to that of how we perform that specific gesture in our everyday lives. However, it has been proven that it is possible to recreate the naturalness of an interaction by changing the mapping of an input that simply borrows an element of the real-life gesture [129]. A correlation exists among the gestures that we perform in such a way that, if our brain can successfully correlate and assigned gesture to be similar in some way to a gesture we usually perform to achieve the same specific task, both these gestures would feel natural. This is closely related to our body schema, which can be altered, mapped and combined between several modalities.

To achieve something that is novel yet with some presence of naturalness or intuitiveness, the newly designed interaction needs to borrow cues from them. In this section, we discuss about how changing the mapping of an input gesture to different parts of the human body presents some novelty in interaction, yet also preserves the naturalness if the gesture can be related to an input in real life.

Correlated mapping refers to the connection between a developed interaction with an interaction that one would perceive as natural or intuitive given the same scenario. The developed interaction refers to how the natural interaction functions, but with a difference; the mapping of the gesture or motion. For example, let's say a user wishes to pick an object on the floor. The natural motion would be to lower the body, reach out the arm, and perform a grasping gesture. What if this entire procedure could be replaced with simply looking at the object and performing a grasping gesture on the hand? By doing so, it borrows some elements of the natural side; paying attention on the object and the grasping motion for example. However, it presents a shortcut due to the mapping of the entire arm motion being just on the hand itself. In this scenario, the naturalness of the interaction is somehow preserved to a certain extent.

1.4. Research Question

To push the boundaries of interaction and develop an input method that can be deemed as more efficient than what we perceive as natural is one of the goals of this thesis. Is there a possibility of the next step of interaction to exist that is more efficient than what we are already used to by ourselves? Does a shortcut exist, and in what way? energy? time? cost? space? Or using a different mapping? If such an interactions exist, can it possibly overwrite our natural interactions? In fact, there has been several works that have touched on the idea of optimizing human motion, though it is more specific such as the trajectory

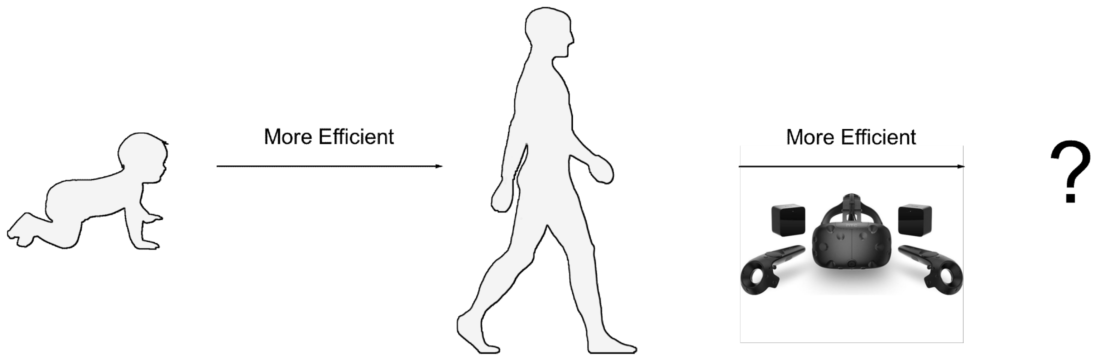

Figure 1.1: Convex Interactions leverage VR to optimize the future of human motions for increased efficiency

planning of the arm movement [155]. This means that researchers to some degree have considered they possible fact that we as humans may be able to adapt to interactions mechanic that are "better". In other words, I would like to discuss the possibility of developing an interaction mechanic that references its natural counterpart, but has been modified in such a way that once we use it, we would prefer to replace the natural counterpart with it instead. In a way, this developed interaction is like a shortcut to a natural input because it aims to be even more intuitive than what we are normally used to.

Convex Interactions explores different kinds of mapped input that can correlate to what we perceive as natural, yet is actually more efficient, making us feel that it is intuitively superior. We simulate these input using VR as shown in Figure 1.1, and physiological signal sensing.To achieve this, we change the mapping of an interaction to other forms of gesture while keeping it within the intimate proxemics space, then further down to peripersonal space, making it more space efficient. The reason why space is an interesting angle to tackle it from is because as mentioned earlier regarding peripersonal space, we in fact have neurons in our brains that directly correlate to the space information around us. By keeping interactions within the small space around us, this isnt just a shortcut in physical motion, but also a shortcut in the neural perception, making us perceive them feeling more intuitive.

Chapter 2

Literature Review

In this chapter, related researchers are explored, starting by the definition of proxemics and peripersonal space, body schema representation, AR/VR mechanics, natural interactions, and related sensing methods.

2.1. Proxemic and Social Space

Edward T. Hall first coined the term "proxemics", which is the study on human space [50] . In his work, he defined four kinds of distances for the human space; intimate distance, personal distance, social distance, and public distance, each with an increasing amount of perceived socially acceptable distance. Quantitatively, each of the defined distances can be measured in metres. Intimate distance is within the range of 0 to 0.45m of the user, personal distance is until 1.2m, social distance is until 3.6m, and public distance is until 7.6m. Referring to our previously defined specifications for Convex Interactions in section 1.7, it was mentioned that it should require no more than 0.45m, which is within the intimate proxemics distance. This perceived social acceptance means that involuntary invasion of intimate and personal space should be avoided. For AR/VR, This presents several design considerations that has very rarely been considered, but is nevertheless necessary as both platforms are evolving to become a more social and ubiquitous platform. It was mentioned in a recent work that technological design needs to support both interaction and sociability [33]. This is a new design approach that is required to facilitate current and future AR/VR imple-

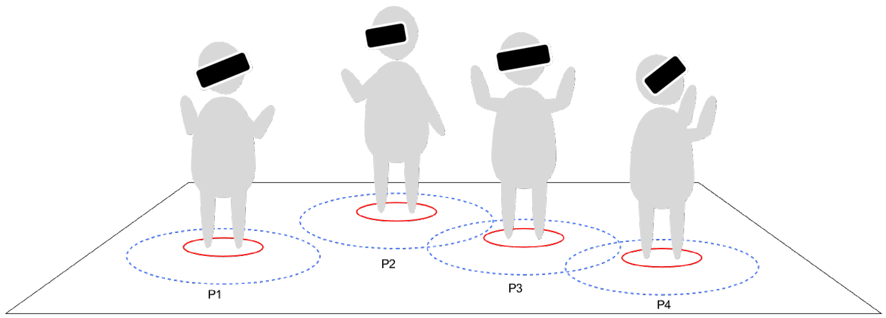

Figure 2.1: Illustration of multiple AR/VR users in a shared space with their intimate and personal space shown

mentations. Establishing a social context is about introducing the constrains of a shared space, such as during a meeting as shown in figure 2.3. In a shared space, depending on context, generally the constrain is the proxemics of other users with relation to our own. If we do not regard this issue, it may lead to possible safety concerns during collision. Meetings like the one shown in Figure 2.3 is generally safe since each participant is seated, but recent AR/VR advancements are moving towards standing experience instead. Therefore, introducing convex interactions is a potential solution to mitigate this issue.

2.2. Spatial Sense from Peripersonal Neurons

With proxemics based on the fixed parameters of space around us, then what of our perception on spatial sense? In this section, we look into the relation of Convex Interactions with neuroscience to understand its connection not just with proxemics, but the understanding of the human brain. Each living being has a protected zone around us; and unlike proxemics, this zone changes in size depending on various factors like our emotion, cultural upbringing and so on. However, this zone is typically within arm reaching distance (equal to proxemics's intimate space and a requirement for Convex Interaction), though it can extend and envelope the tools that we use [84], even virtually [47]. This zone is dubbed as peripersonal space [34]. Most typical studies for peripersonal space is about the placement of object within the zone, though recently researchers have been interested with the notion of the presence of other people nearby and it's effect on it. Peripersonal space is triggered from several sensory stimuli, such as our vision, touch and auditory. Because of this, it can be said that an immersive AR/VR environment can greatly influence our sense of peripersonal space.

Peripersonal neurons are the specific neurons in the brain that actually contributes and encodes this sense of space [48]. What we experience through our senses are encoded into these neurons to create our sense of peripersonal space. Therefore, due to the change of visual stimuli and lack of haptic stimuli in AR/VR, our peripersonal neurons are unable to distinguish and properly encode the environment, thus clouding our spatial sense. This is different with completely removing our visual sense, as it was found that our peripersonal space encoding is still preserved, just with auditory stimulation alone [123]. The issue is from augmenting or replacing our visuals with something else. It can be said that, in terms of neuroscience, Convex Interactionss goal is to re-encode these neurons to reduce intrusiveness and collision in AR/VR space by instead remapping inputs and interactions instead of augmenting vision or introducing haptics. This reduces the proxemic space of interaction and retrains our peripersonal neurons for the virtual space, where we become aware that our input methods take less physical space, thus reducing intrusiveness/collision which are an issue especially for multi-user virtual environments. This applies to single user as well, but the presence of something dynamic, like other users make it harder for the neurons encoding, unlike a static object in space.

2.3. Body Schema Representation

Body schema is about the neural representation of the human body, as opposed to peripersonal space which is the space around the body [55]. However, both are important for us to effectively pilot ourselves through space. It has been proven that, the body schema is malleable in a way that, different functions can be freely mapped around the body and we can easily adapt to them with some training [129]. This is greatly linked to the human cognition's ability to learn and adapt to both new environments and new bodily schema [136]. Therefore, we know that how we perceive our own bodies can be reconfigured in many ways, such as controlling additional arm with our feet [132], or even providing a user

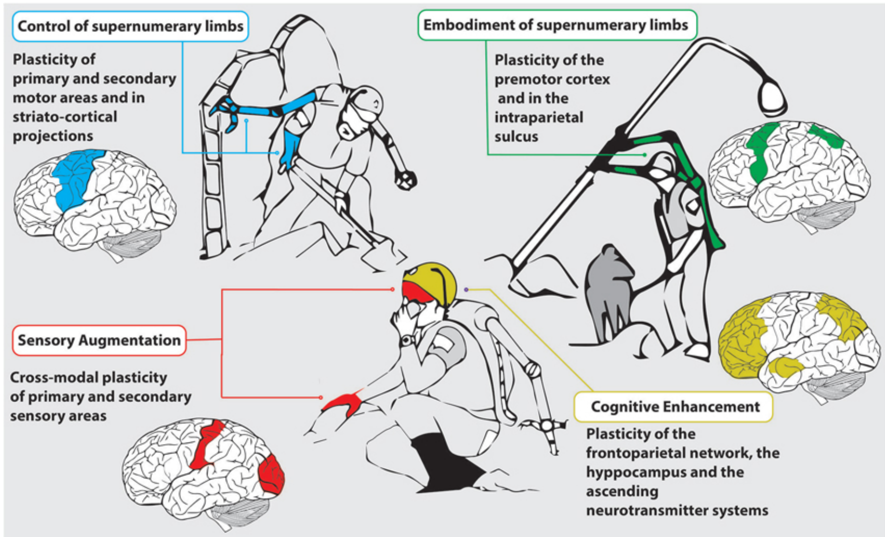

Figure 2.2: Correlated brain regions that are activated depending on the augmentation of the body schema [136]

with multiple points of view at the same time for parallel input [131]. When dealing with the reduction of space consumption in motion, the change in body input mapping will be effected as well. When motion is present, the body schema updates accordingly as well [55].

2.4. Augmented and Virtual Reality Mechanics

Augmented reality (AR) creates many forms of applications [105], but VR holds the advantage in flexibility since everything that is perceived in the environment can be manipulated. To simplify the explanation of VR mechanics, it can be divided into multi-user [18], multi-modal [20], and multi-view [67] for unconventional interaction. An unconventional execution of VR mechanics is World-in-Miniature (WIM), where a scaled down version of the environment is placed in the user hand for ease of navigation and to create a god-like sense of presence [110]. Such a sense of presence is achieved through smart manipulation of the perspective and scale, allowing other forms of applications like interior design [56]. For perspective manipulation, work like Parallel Eyes [67] and JackIn Eyes [68] gives the user a sense of multiple, or another perspective by augmenting the field of vi-

Figure 2.3: Using VR for a collaborative meeting

sion with VR. For multiple users in a shared environment, this concept has been introduced as far back as 1993 when VR was still in its infancy [18].

It is worth noting that despite VR being a platform for novel forms of visualization and input, there are still persisting issues such as the existence of motion sickness, solution for locomotion [63], and lack of a solid haptic feedback system. Since VR is relatively new as a consumer product, there is no rule or defining methods for many of the interactions and user interface mechanics. This allows researchers to continually develop and experiment new interactions, mostly towards and increase of immersion. Walking-in-Place (WIP) has been investigated several times as a solution for VR locomotion due to its naturalness [170]. Other more novel solutions like manipulating the environment directly have also been developed [120]. Another common solution would be to introduce a new peripheral to circumvent existing issues. For locomotion, omni-directional treadmills [32] allows a user to realistically walk in virtual environments, while solutions like a haptic suit [76] introduces the haptic sensations. However, these have not become mainstream simply due to the cost, as well as the idea of introducing new peripherals to a system that already requires many peripherals.

Even though convex interactions introduced in this work can be applied to both AR and VR, similar to some of the aforementioned related work for VR, they are both still fundamentally different in execution and requires separate, related work analysis to properly understand them. The key benefit of AR, and the main difference with VR is that it retains the physical environment, ensuring that the user is still aware of the environment as well as allowing interactions between physical and virtual content. Interactions for multi-view or methods that change the perspective and scale are rarely explored because AR is about accurately superimposing the virtual content with the physical one. Novel interactions in VR are about using new input modalities to manipulate the virtual content, such as using a teach pendant to control the position of the end-effector of a robotic arm for an AR-based simulation [107]. Another interesting mechanic uses voice to activate key points in an AR system [106]. This allows for a more hands-free approach which is useful depending on scenario. Another method solely relies on head-bases gestures to create smooth pursuits interaction [41]. Furthermore, interactions have been developed for single hand use as well, such as scenarios where the occupied hand is used for another peripheral, or a mobile device that renders the virtual content [135]. The free-hand position is recognized using computer vision from the camera of the mobile device, allowing various gesture recognition.

There are other forms of AR too, such as projection-based augmentation so the user interacts with the virtual content similar to smartphone gestures; multi-touch to zoom and pan the content [19]. Another work explored couch-based AR interaction, by equipping a living room-type environment with a depth sensor [86]. The user than can interact with virtual content seen on a TV screen that captures the couch image and displays AR content. This work does not use the first-person view, but rather explores the AR experience in third-person instead. There are also related works that allows the user to project a longer virtual arm to interact with either physical or virtual objects [152], similar to the Go-Go interaction mechanic for VR [117].

However, locomotion mechanics are rarely explored for AR, because this will interfere with how the virtual content interacts with the physical content. For example, if we place a virtual vase on a physical table and virtually navigate to the right side of the vase, then the vase will simply be floating in the air to our left and is not on the table anymore. AR is more commonly used to aid navigation instead, such as providing virtual directional cues for the user [8].

2.5. Natural Interactions

Natural Interactions has been a very controversial, yet extremely important topic of discussion in HCI. The definition of Natural Interaction varies according to context, but it typically means that an interaction is close to that of the physical world counterpart [112]. It could also mean that an interaction mechanic has a high learning rate to quickly transition from novice to mentor, often deemed as a Natural User Interface (NUI). In the end, an interaction is deemed natural because it is defined based on experience: in real life, humans communicate not just vocally, but through gestures, expressions, and movements, as we discover the world through manipulation of the physical matter around us [157]. Other related works also link gestures towards natural interactions [24,163]. It has been proven that gestures directly make navigation around an interface or as an input for digital games more engaging [13]. This is because involvement of the body motion creates a strong effective experience and a sense of presence. Gestures, interestingly, exists in many forms with some of them subtler than others. One of them exists in the form of physiological signals, which can be used as a form of natural interaction [89]. However, Donald A. Norman argued that user interfaces that usually claim to be natural are in fact not natural, because given time, humans are actually somehow able to adapt to that interface, unless it inherently causes us physical harm like motion sickness [94]. He did agree though, that gestures play an extremely important role for us to understand exactly what is natural. In this work, we strive to challenge the concept of natural interactions as well, by introducing interaction mechanics that could possibly initially perceived as being unnatural.

2.6. Sensing Methods

Each of the aforementioned wearables, besides notification-based, explores various sensing methods to read different kinds of data from us, usually by interpreting some kind of subtle gesture. These gestures recognition systems are also present in AR/VR, such as gestures based on head rotation and movement [66, 148].

Physiological sensing, on the other hand, relies on more unconventional sensors, though it is not wrong to say that gestures are a form of physiological signal as well. For example, Brain Computer Interface (BCI) have been used for direct manipulation in VR for navigation [46]. Electromyography (EMG) is another form of sensing for muscle activity that has been used for various medical applications like muscular rehabilitation, muscular disease, prosthetic control, and robotic exoskeletons [9, 52, 62, 72, 90]. It is also highly promising given its discrete nature of activation [90].

Another popular sensing methodology is eye tracking [161]. It allows for direct manipulation in two axes, and if coupled with VR, provides us with a subtle approach since the users eyes are completely concealed by the HMD. This makes eye tracking ideal for most VR interactions and will very likely be integrated in the next few VR iterations [45]. One of the more important factors for eye tracking is the right calibration, as it is heavily dependent on the user. In the past, neural networks [39, 116], heuristic filtering [143] and parameterized self-organizing map [114] has been used for calibration methods. However, integrating eye trackers with current consumer VR have proven to be tricky, and available solutions are being sold at over $10,000 [140], limiting its uses to only researchers. For AR, even though it is possible to obtain eye trackers that can be mounted on the Hololens [158], the cost combination of both the Hololens and eye trackers approaches $10,000 as well.

There are many other physiological sensing methods that exist, some of which will be covered more extensively in this study depending on sensor availability [171]. Comparatively, physiological sensing solutions are subtler, though less immersive, when compared to gesture-based solutions. Of course, these depend on the context and application.

2.7. Summary

In this chapter, we highlighted some previous related work in the concept of space, its relation with body schema, the contributions of AR and VR, the definition of natural interactions, as well as various sensing methodologies to achieve it. From the many discussed interaction mechanics, I would like to delve deeper into interactions that are more fundamental; the things we as humans do everyday, which is one of the key parameters for Convex Interactions.

Next, I will delve deeper into the definition of Convex Interactions, the analogy behind its name, and how it is used for shortcutting interactions for increased efficiency through explorations of several defined parameters.

Chapter 3

Convex Interactions

In this chapter, I will discuss about Convex Interactions, the work coined in this thesis. I will define it as well as explain the reason behind the naming of the interaction mechanic.

Convex Interactions are interaction mechanics that utilize our proxemic and peripersonal space sense to shortcut interactions, both spatially and through exploration of bodily mapping, to create a more efficient and intuitively superior form of interaction than what we are used to. Essentially, it uses what we can learn in the field of human-computer interaction (HCI) to improve human interaction (HI) itself. The term 'convex' was used to describe this because it represents the concept of light focusing in optics. A convex lens focuses light from an external source towards a single point in space, as opposed to a concave lens that disperses light in multiple directions. We use this analogy to describe the proposed interaction mechanics, illustrating how interactions in virtual environments are converted to be more focused around the user's proxemic distance, or minimizing interactions proxemically as shown in Figure 3.1.

For the more mathematically inclined, it can also be compared to a convex set of numbers as shown in Figure 3.2. A set of numbers is considered as a convex set if, given any two points in that set, the line joining them lies entirely in the set. Therefore, the convex set of numbers can be considered as Convex Interaction's zone of interaction, whereas the line between the two points can be considered as the line of interaction. If an interaction takes a wide amount of space, it should be scaled down to be within the intimate proxemics. If an interaction was already

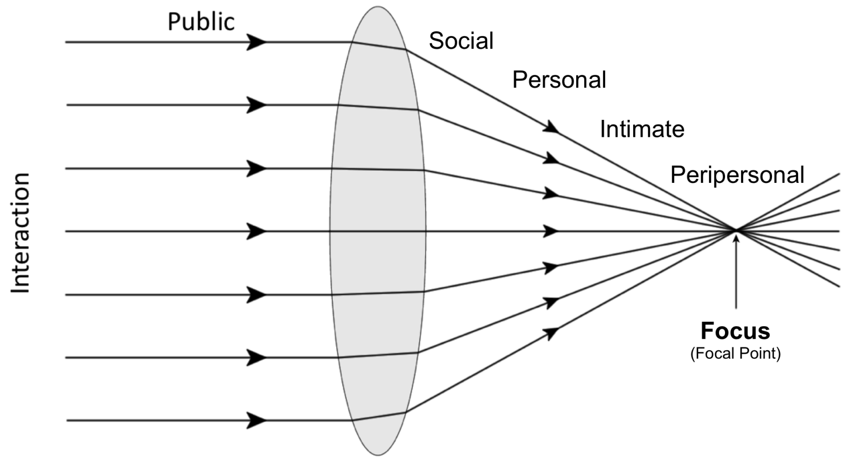

Figure 3.1: An illustration of Convex Interactions' analogy with proxemics

within the intimate space, it should be scaled down to within the peripersonal space of the user.

The simplest way to explain convex interaction is that, in terms of mapping, it deals with a different mapping system for input that is not 1 to 1 to the human motion. For example, if we reach out to grab a virtual object, currently this would mean an identical motion of reaching out to grab a physical object if both the physical and virtual object exist at the same point in space. Convexing this would mean reducing the amount of space used by remapping this function to another limb or shortcutting the process by removing some procedures. The trigger to grab the object is due to a signal from the human body, as opposed to the mechanical trigger of an external activation like a button. Mapping of input greatly effects or sense of presence and immersion if done correctly, evident by VR itself which is essentially a mapping of camera control to head rotation. In this thesis, we also explore other types of gestures as well as physiological signals.

However, determining the specifications of what qualifies as a Convex Interaction is still not a simple matter, though right now we can hypothesize the approach. Firstly, we need to understand how shortcutting an interaction will effect the users perception and performance. Then, we need to experiment on the possibility of changing this mapping or function to another limb to conserve spatial use.

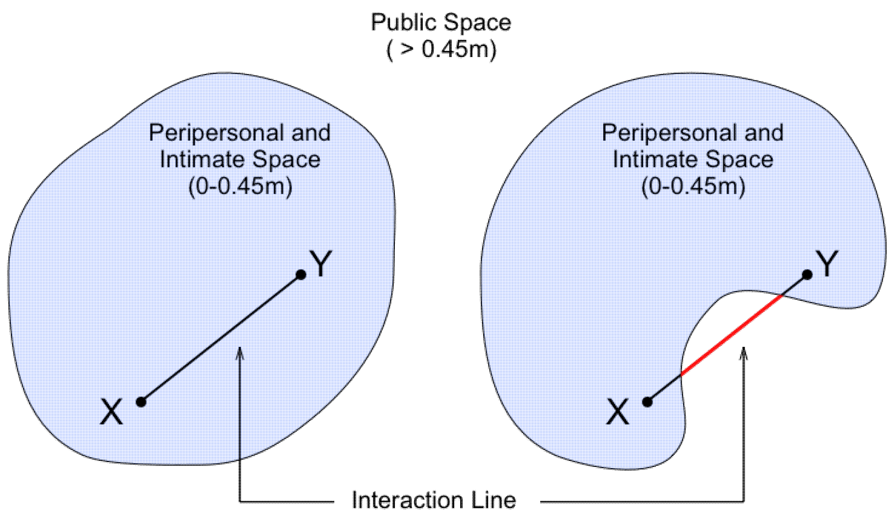

Figure 3.2: An alternative interpretation of Convex Interactions' analogy with respect to convex sets

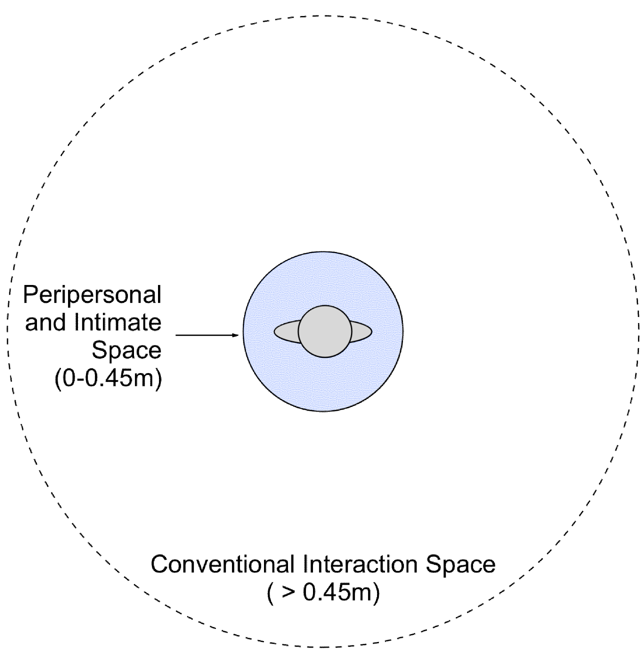

Figure 3.3: The maximum amount of space used for Convex Interactions derived from proxemics compared with conventional interaction space

Trying to deal with every single form of interaction that currently exist for a human is almost impossible; therefore, I look into 2 generic interactions that I believe covers our spatial reach most. The first being locomotion, and the other being selection and activation. Locomotion, as I have discussed since the Introduction chapter, is a suitable example because it is an interaction mechanic with space that we know of since the moment we are born, and has evolved from us from crawling to walking. Therefore, understanding the next locomotion mechanic that could potentially be more efficient than walking serves as a suitable progress for Convex Interactions.

3.1. Parameters

I look into several parameters that will influence my design for Convex Interactions. Based on what was discussed earlier, I have established that the first key component in designing these new forms of interaction is the space constrain. However, changing this depends on the interaction type and would also effect how the mapping is conducted, the motions involved, and even the direct output of the interaction. Therefore, the five key components are:

- Space type
- Interaction type
- Mapping Type
- Motion Type
- Output Type

These parameters will be explained below. Our full approach in the next chapter are designed based on the parameters listed here.

3.1.1 Space type

As I have established before, the first parameter would be the space type for the interaction being performed. This is divided into intimate space, or peripersonal space. Intimate space is a radium of 0.45m around the user, meaning interactions that require the user to extend his/her limbs within this distance falls into this category. Several limbs can also be moved at the same time or procedurally. In terms of vertical space, there is no constrain. Peripersonal space governs interactions with no obvious output when observed from the outside. This includes eye movements, muscle contraction, tongue movement, etc. Limb movements are possible as well, though only rotation of the limb about its axis.

3.1.2 Interaction Type

Since there are too many forms of interactions that a human can perform, this parameter divides interactions into two categories; serial and parallel interactions. Serial means that, to reach a desired output, the user performs a series of movements one after another in a procedural manner. Selection and activation is a form of serial interaction, where to achieve the desire of grabbing an object in space, the user must first look at the object, reach out an arm, touch the object, and grasp the fingers to grab it. As for parallel, this interaction type means that several motions are involved at the same time to produce the desired output. Locomotion falls into this category, where the user's intention is to propel himself/herself forward. In this case, the motions of the left leg, right leg, left arm, and right arm are all activated at the same time to achieve this.

3.1.3 Mapping Type

Mapping type is related to the correlation of body schema and the changes and modification of it when designing a new input for a desired output depending on the interaction type. The modification is divided into two type; shortening and rearranging. Shortening means removing additional procedures required for a desired output of an interaction, such as reducing the motion of walking to that of only the feet or hand movement for locomotion. Rearranging means, in a given interaction type, the output of any interaction type can be paired with any input motion, such as mapping a locomotion output to that of the neck motion.

3.1.4 Motion Type

Motion type refers to the motion of the body parts required to achieve the desired output, measured through the physiological signals of the user. This is simply divided into the same motion type, or different motion type. The same motion type refers to, regardless of the mapped body schema, the motion itself remain unchanged. For example, the same motion of the arm swing is preserved for locomotion, even if it has been modified where leg motion is not required for locomotion. A different motion type means, for a desired output, the motion itself is different from the expected motion. For example, grabbing an object requires the motion of moving the entire arm, but the motion itself can be changed to a simple flex of the arm muscle.

3.1.5 Output type

The output type governs the expected output of an interaction, as has been divided into linear or non-linear. A linear output type means that the output is exactly as expected, such as a the locomotion interaction would expectedly create an output of moving forward. If the output was non-linear, then it would be different from what a typical human would expect it to be. In the locomotion example, perhaps the output becomes a change in elevation instead, of a different movement trajectory as opposed to the norms.

3.2. Research Contribution

This research does not focus on the development of new wearables or hardware for the interaction, but rather on understanding the right gesture and sensing methodology to push human interaction to the next stage. The contributions can be summarized as the following:

- To explore the possibility of beyond natural interactions as the next step of human interaction evolution.
- To understand how spatial constrains contribute to more effective motions and gestures with the aid of VR

- To explore various correlated mapping input using physiological signals
- To suggest scenarios and applications where Convex Interactions may contribute, within the field of VR and beyond

3.3. Summary

To summarize, Convex Interactions is just the first step into understanding the evolution of human interaction. At this point of reading, it is currently just an hypothesized interaction mechanic that can theoretically improve natural interactions. In the next chapter, I will proceed through my approach in understanding Convex Interactions more through the development of new interaction mechanics using VR as a tool.

Chapter 4

Approach

This section underlines my approach towards understanding Convex Interactions. Previously, we defined what is an efficient interaction, and understood that based on related work, we can see that 1) VR is a suitable tool or platform to emulate human motion and vision due to its sensory immersion, 2) the motion inputs in VR can be measured from the human physiological signals, 3) the mapping of the body schema can influence our perception of motion, and that 4) the perception of space exists both in the physical environment as well as on a neural level.

However, before claiming that a certain motion or gesture is able to overwrite what feels natural to us, detailed studies still needs to be conducted to explore each of this aspect properly. This section uses VR as the main medium to understand Convex Interactions, with each approach aiming to answer a specific research question.

4.1. Workflow

As it is not possible at this point to actually look into every single available human interaction, my approach is divided into two main interactions that is most spatially consuming. The first interaction is locomotion, which is a user's motion to induce his/her change of position in space, and the second interaction is selection and activation, which are pick and place activities in general, or interactions with nearby objects in space. This section will look into five stages of research on development, with each of them catering to a research question. The first three are on the locomotion scenario, whereas the last two delves into selection and activation.

The first approach is to answer the following; in a locomotion scenario, several motions and gestures are involved. Given that, I looked into the approach of increasing efficiency by reducing space of interaction to within the user's intimate space, reducing the number of motions involved, as well as reducing energy consumption while preserving the naturalness of the motion. With this modifications, will the user find it preferable over ordinary locomotion? For this study, we do not change the mapping of locomotion, but instead simply perform a shortcut to the existing motions involved.

The second approach is to answer the following; based on the earlier results, we now look into the correlated motions with regards to locomotion. We now directly map precise control of the feet motion's output onto the hands. This approach also reduces the space of interaction to that of the intimate space, but the motion is programmed in such a way to be one-to-one with the user's motion. By doing this, are we able to maintain, or even increase the accuracy of user?

The next approach answers the following and is the last locomotion scenario; we now take a more drastic approach to space consumption in the motion by reducing it to within the user's peripersonal space. As defined earlier, peripersonal space changes when the user moves or grabs an object. Therefore, interaction within this space means that the motion generally involves only rotation of limbs, and/or movements that do not extend outwards such as eye movement. We also further push this by mapping the actual output of locomotion from linear to rotational to understand user's adaptation. With this drastic modification, can the user still adapt to a direct change of output and space constrain in such a way that it can even potentially overwrite our preference?

The next approach moves to selection and activation; this scenario differentiates in a way that, this particular form of interaction is more procedural-based, as opposed to multi-input like locomotion. For selection and activation, we perform a motion, followed by the next, to achieve the desired output. Therefore, the shortcut approach for efficiency in terms of space and number of involved motion for this interaction is studied here. According to what we found in the previous 3 results, can this be applied to other forms of interaction?

The final approach looks into just activation; this can potentially be difficult to answer, because this can possibly be substituted with a button input. In this case, we directly compare button-based input with physiological signal input. In such a case, the physiological signal input needs to be as usable as a button

4.2. Sensing Methods

In this section, we discuss on the different sensing methodologies that will be investigated to facilitate Convex Interactions. The key objective in this particular section of study is to identify gesture input methods through the sensing of our physiological signals. As mentioned previously, conventional input is achieved with buttons on a gamepad or through full motion.

It is important to note that this study will place several limitations on sensing methods depending on two key factors; implicit and explicit interactions. We focus more on explicit interactions so that various input modalities can be studied as an alternative over conventional input like buttons and motions in a VR environment. Among the various forms of explicit sensing used here is eye tracking, muscle sensing, and gesture recognition, which will be the key emphasis of this study. Implicit sensing generally cannot be controlled directly by humans and acts more like a reflection to our physiological state, such as heart rate, blood pressure, and temperature. Since there are plenty of physiological signals in the human body, it is not possible to cover every forms of interactions by all the signals. We also primarily look into sensing methods that can augment the HMD itself so that it remains hands-free and clutter-free for minimal intrusiveness especially for intimate proxemics. For the arm swing work, arm gestures use the readily available controllers to track the movement. These are not readily available for AR devices at this point of writing, therefore VR motion controllers are used. To briefly cover each interactions, arm-based input investigates the use of arm swinging motion to navigate in a virtual environment. Forearm movement uses intermittent pinch gestures for accurate navigation in virtual space. Towards more subtle interactions, we investigated the use of head movement and facial muscle gesture to facilitate sports spectation. We then further investigated the combination of head movement with eye gaze tracking to navigate 360-degree-video environment, as well as head movement for a more directed approach. Next, we look at the multimodal approach of eye gaze tracking with arm muscle contraction as an alternative hands-free input modality. Finally, we look at eye gaze tracking only for the most minimal movement as an input.

4.3. Convexed Locomotion: Arm-based Gestures

Our first look at convex interaction is based on arm gestures which are sensed using the conventional motion controllers. The two developed interactions; arm swing and PinchMove, will be explored in this section.

4.3.1 Arm Swing

This section is adapted from a full paper presented at the ACM International Conference on Mobile and Ubiquitous Multimedia (MUM 2017) [100]. The paper was co-authored by myself and Kai Kunze. Arm Swing is a locomotion mechanic that falls into the intimate interaction space. Since locomotion is a parallel-type interaction, it shortens the required interaction by removing the primary motion (feet movement) and emphasizing the secondary motion (arm swing). However, the motion type was not changed as arm swinging was still used, and the output is linear where the user moves as to be expected.

Navigating virtual spaces in VR often causes motion sickness. It might be a critical barrier to use VR as effective rehabilitation and training tool. One attempt to overcome this sickness is to simulate locomotion [79]. Current research implementation related to locomotion in VR such as walking in place (WIP) methods aim to create a more realistic sensation of walking, thus negating motion sickness by avoiding contradiction with the body's sense of balance and spatial orientation [151]. However, as VR usage becomes more mainstream, WIP suffers from several issues, mainly 1) jogging in place causes potential drifting and uses up up space, and 2) it becomes tiring after a slightly extended period of usage, unless it was designed for energy consumption like sports simulation in the first place. This creates a barrier for some, such as elderly consumers or just the general public who wants to use VR more socially. Unlike foot or head-based WIP which enables locomotion through foot motion or head bobbing, using arm swing enables users to navigate a virtual or augmented environment simply through arm swing, allowing for a more socially acceptable interaction while preserving the realism of walking that is natural to human gait, as well as consuming less energy. The user simply needs to swing their arms in a natural movement as they do when walking, providing them the freedom to look around without affecting the walking direction. No feet movement is required, making any additional sensors or 3rd party peripherals unnecessary. Users also do not need to deliberately bob their head to allow a better immersion and focus in the virtual scene during locomotion.

The implementation of arm swing gesture pursues 4 major goals: to develop a virtual space locomotion solution that feels natural, preserves immersion while using less space, ensure that users can easily utilize the system without additional devices, and evaluate users' feedback on the energy consumption, ease of use and immersion factor of arm swinging compared to existing WIP methods such as VR-Step [149]. Other locomotion solutions in VR like the omni-directional treadmill [32], brain interface, etc. has been developed towards the similar goal of immersion, yet they involve hardware that are not easily accessible, too costly for the average consumer, or simply adds to the number of peripherals for VR systems that are already plentiful by default, making these solutions viable only in research labs or specific applications. This makes software approaches like arm swing gesture preferable. Furthermore, with the inclusion of hand position tracking controllers for the current and future generation of VR solutions, users do not need 3rd party tracking devices. Arm swing was never a necessity for human locomotion, however this motion feels natural for humans when walking or running [5], and simulating this gesture in VR has the potential to induce the sense of navigation while maintaining immersion.

Navigating a virtual space highly depends on the kind of system being used. For example, mobile systems like the Google Cardboard and Gear VR does not have any physical controls, making these choices rather limiting when it comes to any form of interaction in VR space, since the user essentially only has gaze-based interaction. The Oculus Rift DK1 and DK2 relies primarily on a game-pad controller for both interaction and navigation, while the current generation of VR devices, namely HTC Vive and the new Oculus Rift has controllers that are tracked in physical space. The AR space is even more scarce, with the Hololens having no hand tracking and mobile solutions like ARCore and ARKit still renders virtual content on the phone display only. With such variability in VR systems, various researches have been conducted to determine the best possible locomotion method in VR space. One of the biggest advantage offered by the HTC Vive is the room-scale experience, where the user can physically walk around within a confined physical space thus improving the sense of presence [139]. The Vive uses a Chaperone bounding system for the user to see the physical boundaries in virtual space, so the boundary does not need to be integrated by developers [26]. This allowance for actual walking is in line with a human's psychological requirement that physical movement is more important than a rich visual scene when it comes to locomotion [126]. It was found that both transitional and rotational body movement helps for efficient navigation [127], though another research showed that physical rotation is sufficient for actual walking, implying that immersive locomotion can be achieved without the required physical space [124].

More natural interactions where the user simply performs a jogging action on a spot to navigate have been gaining popularity because spatial information is the same as the real environment, therefore humans require the correct motion to adapt to any change in the virtual world [113]. One of the more popular methods is called walking-in-place (WIP), where the user simply performs a jogging action on a spot to navigate. VR-STEP was one of the newly developed method aimed for mobile VR that leverages inertial sensors in the smartphone to provide the user with a realistic method of locomotion [149]. However, this system only caters for mobile VR and AR. Another WIP implementation is by using a Wii balance board [164]. Since the board has pressure sensors, it was relatively straight forward to use it as a locomotion device for virtual environments. This proved reliable and that the Wii board can be easily obtained, though users still need to rely on this additional peripheral to couple with the already arguably cumbersome HMD and controllers for a VR setup. The same can be said about another work that uses the Microsoft Kinect to detect walking [167]. The depth sensors in Kinect allows accurate skeletal tracking of the user by measuring the angle between the hip, knee and ankles. Compared with the Wii board, gesture based recognition means that the user is at least not in physical contact with the peripheral, preserving relatively more immersion. Other approaches for WIP are by attaching calibrated sensors on the legs and calf [169], however, WIP methods tend to be more tiring and a continuous jogging motion may cause perspiration and to a more serious degree, nausea. This is the reason why VR applications rarely use head bob, which is a way to show that the virtual character is moving by bobbing the camera [153]. Furthermore, WIP can only be used while standing, whereas arm swinging can be achieved at sitting or standing scenarios.

Arguably, the best method for locomotion when it comes to immersion is by using an Omni-directional treadmill (ODT) [32]. It was initially developed for the U.S. Army's Dismounted Infantry Training Program and it allows the user to realistically perform walking motions, yet still remain at a single spot. The main disadvantage is that a custom treadmill like the ODT is surely too costly for average consumers, if even accessible at all.

There are also more unconventional methods for VR locomotion, such as a flight based locomotion by manipulating the sense of scale [110], rotating the virtual environment [120] or simply using head angle [148]. One of the more unique methods of navigation is by using brain-computer interface [46]. In this method, electrodes are attached to the user's head to obtain electroencephalogram (EEG) signals that are used as input values for the virtual environment. However, brain interface tends to have inherent issues such as the presence of noisy data, as well as it not being accessible for the average consumer.

With this current generation of VR systems, most developers rely on on-rails sequences, controller-based, or teleportation-based navigation. On-rails simply mean that the user is not given the freedom to walk around and is confined to a fixed rail that usually consistently moves which can be seen on games like London Heist: The Getaway and Walking Dead. Blink teleportation is a relatively new method proposed by Cloudhead Games [27] for VR where the user simply points and teleports to the designated spot. Lastly, controller navigation using a gamepad, normally maps the left analog stick to locomotion and the right stick to head movement. For these three methods respectively, the user has no freedom of movements, teleportation is not realistic, and gamepad controls induces motion sickness which is more prominent for VR.

The closest to our proposed method is a research by Mc Cullough et al. and

Wilson et al. [85, 166]. Both use an arm swing method similar to the one proposed. However, they require additional hardware (the myo-arm band). The band has to be adjusted in a specific angle (susceptible to shifts and not placement-independent). They are also using simply angle changes on the upper arm or velocity to detect the walking speed, not the trajectory of the walking direction. The user in VR walks in the direction of their head orientation not in the direction of the arm swing. Our approach in comparison works without additional hardware, is sensor-shift/orientation robust and uses arm swing trajectory for walking direction. The user can look in any direction while walking.

Using the arm swing movement is often only limited to bipedal locomotion [5], and is a rather interesting proposal as studies have shown that arm swing reduces the moment about the vertical axis of the foot while walking [108]. This means that a relationship does exist between arm swing and foot reaction, despite it not being necessary while walking. In fact, total energy consumption with arm swing is lower than without during walking even though energy is consumed for arm movement, therefore overall reducing the cost of walking [29]. Because of these traits, researchers have utilized arm swing for walking rehabilitation [74] and robotics [28].

Implementation

The mechanics of using arm swing relies solely on the users' arm swing motion, akin to normal walking gait. No form of additional pressure sensors is required for the feet movement tracking, or that deliberate head bobbing is necessary. It is important to note that the facing direction and walking direction should be different as well. This means that the user should be able to continuously walk in the direction their body is facing, yet still may freely look around. Furthermore, in an effort to preserve space used for arm swinging, the algorithm does not depend on how far the arm is swung, rather the relative positions of the arms. A solution for this will be further explained in the next few paragraphs. Since arm swinging is achieved using purely motion recognition, no buttons are actually required, and can be reserved for other forms of interaction if the scenario desires. This is shown in the user study, where a simple task is assigned for the participants using the trigger buttons. ArmSwingVR was developed using the HTC Vive because as the moment this work was conducted, only the Vive comes with positional tracking controllers. Therefore, only the Vive controllers and headset are necessary for the user to fully use the proposed method. The tracking space used for development is 1.6m x 3.1m, though for the user study, the tracking space will be maximized (4.6m x 4.6m). The entire system was built using the Unity development environment for seamless integration with the SteamVR plugin. C# was used as the primary coding language. For a smooth VR experience, a desktop Windows PC equipped with a Core i7-6700 processor and an Nvidia GTX 980 graphic card was used which is above the recommended specifications.

$$\overrightarrow{AB} = \overrightarrow{B} - \overrightarrow{A}$$
$$\hat{A} = \frac{\overrightarrow{A}}{|\overrightarrow{A}|}$$
$$\overrightarrow{AB_z} = \overrightarrow{AB}.\hat{A}_z$$

Finite state machine was used to enable the system to recognize the position of both controllers relative to the position of the HMD [148]. The relative positional vectors for the controllers and the HMD must first be determined so that the system knows which of the objects are in front of the other regardless of the facing direction. We are only interested on the forward vector component because the height and side vectors are not important for arm swing. For the relative positions between controllers, relative vector AB is found by subtracting vector A with B. The transform vector of A is then normalized and the dot product between AB and the normalized A is used to find the forward vector component. This procedure is repeated to find the relative position between the HMD and the left controller (called vector AC), as well as between the HMD and the right controller (called vector BC).

Six states were constructed for the motion detection, which are *Idle*, *RightFront*, *LeftFront*, *RightFront2*, *LeftFront2*, and *Walk*. Figure 1 illustrates the algorithm that connect each of these states together through a series of decision making. The *Idle* state is the default state of which the user is standing or interacting with the virtual environment. In this state, no velocity is induced and the system checks the positions of the controllers. If AB is positive, BC is positive and AC is negative, this means that the right controller's position is in front of the HMD whereas the left controller is behind. This changes the state to *RightFront*. A similar decision process is made for *LeftFront*. In this next tier of state with *RightFront* as an example, we then check the duration t of which we are in the state. If t is within 1 second and the position remains the same, the state reverts back to *Idle*.

Otherwise, if the position switches such that the left controller is now front, *LeftFront2* is then initialized. *RightFront2* and *LeftFront2* is a safety state layer that ensures that the user does not accidentally walk, and will only do so on purpose. Similarly, the duration is checked again, and if the arm positions switches once more, we finally enter the *Walk* state. The actual walking motion is conducted in this state. Velocity is induced based on the walking direction. The walking direction is not the same as facing direction, which is common in most first-person view camera applications. In real life, we may walk in a direction yet face another. To achieve this, the walking direction has to depend on the forward vectors of the controllers. The resulting walking vector D is found by summing vector A and B. Figure 4.2 depicts these vectors from a top-down view. Linear interpolation is then used to smooth the change of vector D when the controllers are constantly swaying back and forth. Furthermore, only the rotation about y-axis of the vector is important since we just wish to know which way it is facing and not if the vector is facing downwards or tilting.

It is important to use Quaternion rotations for this because a controller that faces downwards, which is common when walking, will be subjected to gimbal lock if Euler rotations were used. If the user wishes to stop walking, several conditions need to be met. One of the most important parameters is finding the speed, s of the controller. If the speed of the controllers approach zero, it is highly likely that the user has stopped swinging their arms. However, the speed also approaches zero at the amplitude or maximum swing of the arms. To solve this, we created two conditions: walking is halted when the controllers' velocity approaches zero and are close to the HMD at the z-axis, or when the controllers speed remains close to zero after a period of time. The first condition uses the vector AC and BC computed previously to determine if the user has stopped any motion and is standing still. The second condition uses a Coroutine that delays the next

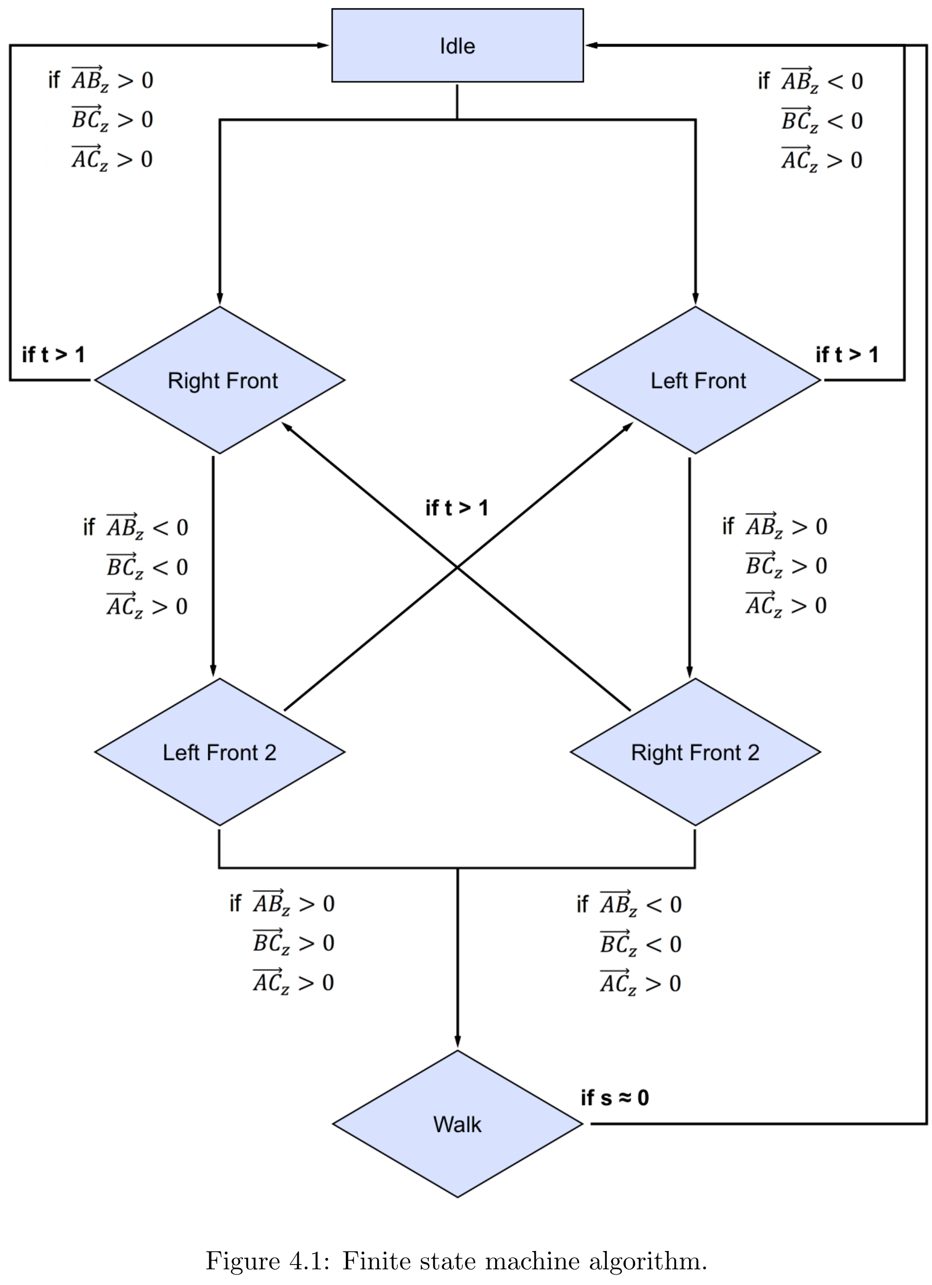

Figure 4.1: Finite state machine algorithm.

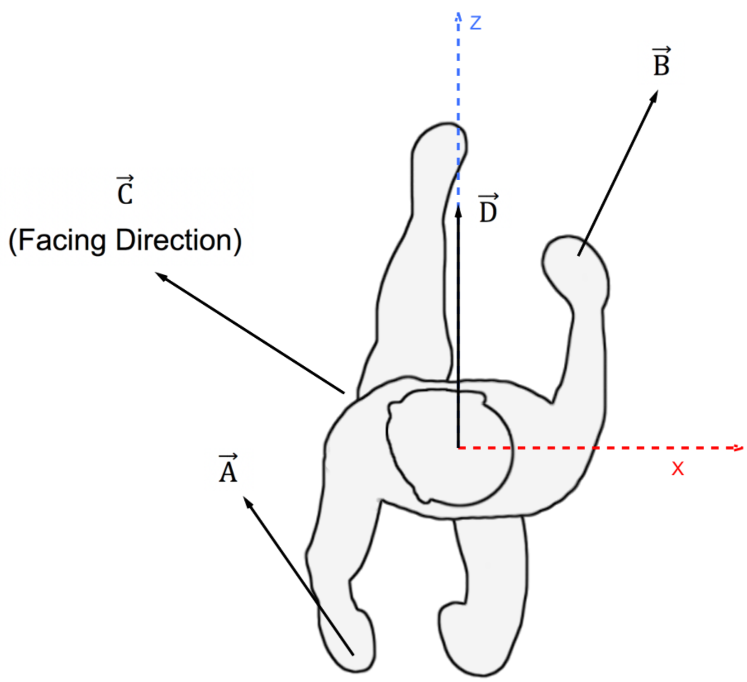

Figure 4.2: Top-down view of the directional vectors.

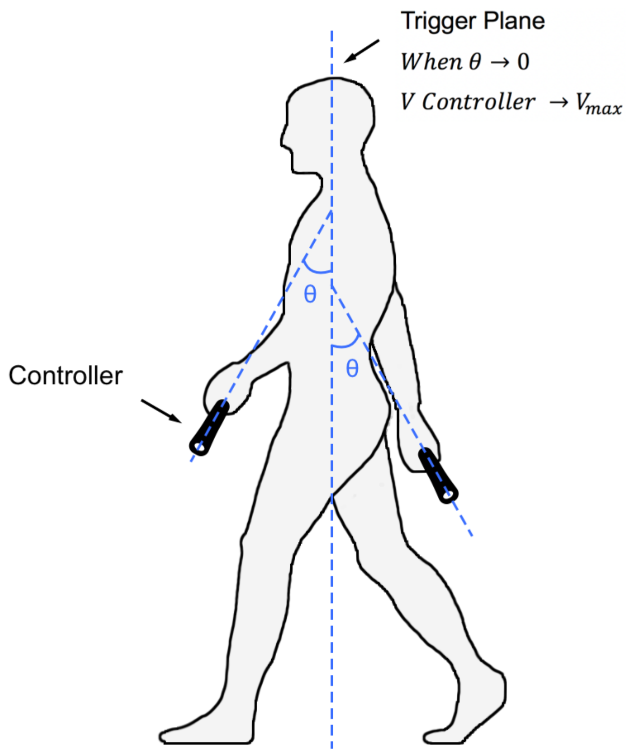

Figure 4.3: Placement of the trigger plane for determining the point.

checkpoint by 2 seconds. If the controllers are still relatively static after that period of time, the state finally reverts back to *Idle*.

The user can control the walking speed depending on the speed of arm swing. However, as mentioned earlier, the speed approaches zero at the maximum or minimum swing point. This causes a jerky movement as the velocity of the movement constantly fluctuates between zero and the current arm swing velocity. Therefore, we created a trigger plane placed on the HMD as shown in Figure 4.3 and only take the velocity of the arms at the point of collision with the trigger plane to ensure that only the maximum velocity value is used.

The resulting system is a relatively solid locomotion method with low accidental stops and walks, while allowing the user freedom to look around and control their walking speed without the push of a button.

User Study

The user study for arm swinging is a comparison with WIP based on 4 factors; performance, motion sickness, energy consumption and immersion. The user study focuses on a direct comparison between the mechanics present in VR-Step (used to represent WIP-based solutions) with our method, both which are easily accessible for consumers and are software-based solutions. Furthermore, since this study is aimed towards standing VR experiences, to some degree, the users are able to physically walk around a fixed amount of space due to the Vive's tracking. Our implementation of VR-Step was based on the HMD's spatial tracking data. The user firstly needs to press both the grip buttons while standing still to calibrate the height data. This creates a trigger collider above the HMD that can only be triggered when the WIP state is activated. This is as similar to VR-Step in which a distinct jogging motion is required [149]. For this method, a slightly modified state machine algorithm was used, composed of five states; *Idle*, *Transition*, *Triggered*, *Walk*, and *Walk2*. After determining the height of the collider, the system checks if the user's head enters the collider. If it does, the *Transition* state activates. This state acts as a transitional phase or safety measure to determine if the user desires to walk or was simply performing other forms of interaction. In this state, the system checks the user's head's location within 0.5 seconds. If the head has exited the collider, the *Step* phase is initialized. Otherwise, it goes to the *Triggered* state. This state activates when the user's head has been in the collider for a period of time. If the head exits the trigger again within 0.5 seconds, *Step* is registered. Otherwise, it returns to the *Idle* state. In the Step state, velocity is induced to the rigid body, creating a forward motion at the facing direction. Unlike arm swing, facing direction and walking direction are not independent. Maintaining this velocity depends on the user's capabilities to alternate the head position from entering and exiting the collider, i.e. a head bobbing motion. Therefore, *Step2* was created for when the head enters back the collider while in the *Step* state. *Step2* then reverts back to *Step* when the head exits the collider again, for a continues induced velocity. If none of this conditions are met, the state reverts back to *Transition* to check again, and no more velocity is induced. Each participant is given about 5 minutes to familiarize themselves with the locomotion controls for both methods before the study is initialized.

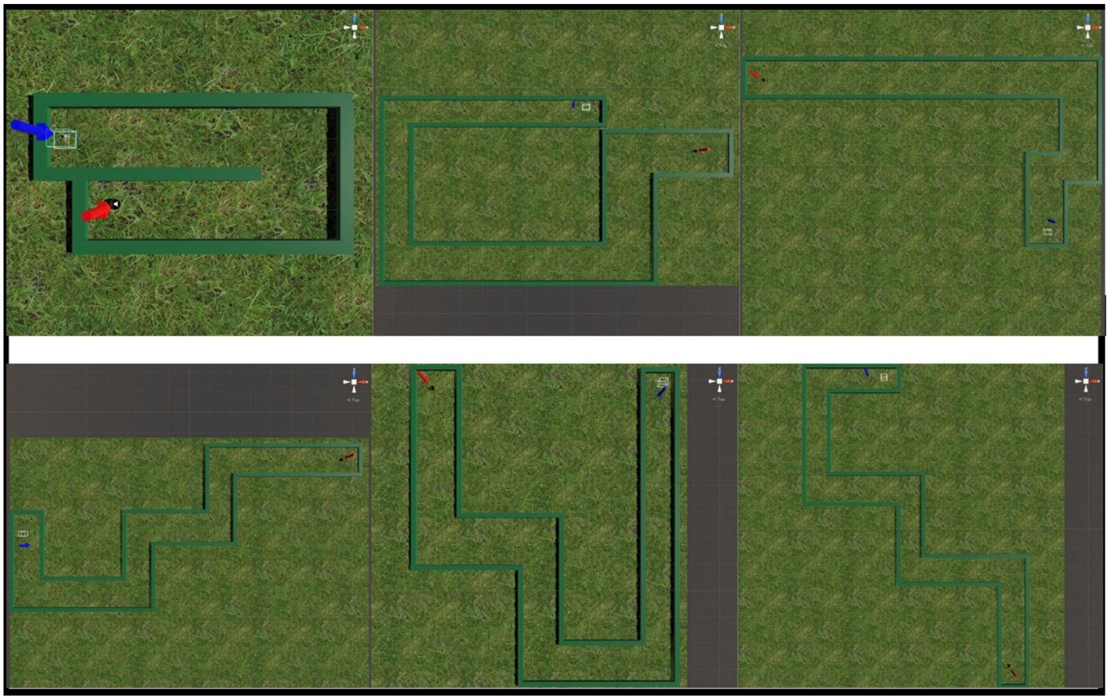

Figure 4.4: Virtual routes for the user study.

To ensure that the speed of movement of the participant for both the locomotion methods are the same, the previously mentioned feature regarding controllable locomotion speed is disabled. We obtain the immersion and Simulator Sickness data through the Presence Questionnaire and Simulator Sickness Questionnaire (SSQ) [69]. To determine its effectiveness, the participant is required to navigate a series of routes shown in Figure 4.4 while picking up virtual balls that they find using the trigger button.

These balls must then be placed into a virtual basket located further down the route, where it counts for 1 point per ball. After each score, the next route appears and the participant must repeat the process. Figure 4.5 shows a giant arrow hovering over both the ball and the basket so that their positions are known. The routes were designed in a way that forces the participants to navigate in all four directions. They are also straightforward to eliminate any requirement for the participant's spatial awareness. This task runs for 15 minutes per participant, for each of the method. Even though extended VR sessions are not advisable [95], VR is improving every day and full story-based 3D games are being developed for

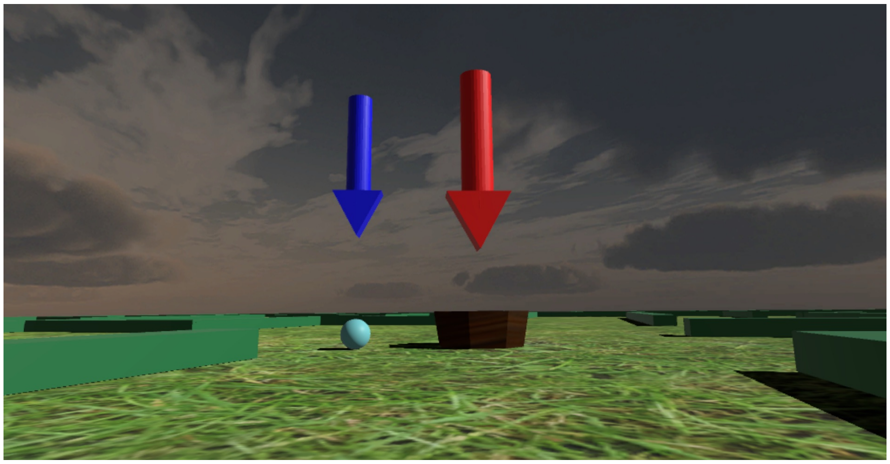

Figure 4.5: View of the participant, where the blue arrow is the location of the ball and red arrow is the location of the basket.

VR as of this moment, therefore we deemed it necessary for participants to spend 15 minutes for each method to determine the outcome and energy expenditure.

At the end of each locomotion experiment, the participants are required to complete the SSQ to determine their feedback on any induced motion sickness [63]. To determine the immersion and sense of realism, the presence questionnaire [168] was deemed suitable because it covers a wide area of applications including locomotion in VR. We chose to exclude the sound and haptics-based question as they are not related to the current study. To determine energy consumption and workload, the NASA task load and heart rate monitoring is used [43]. For the quantitative analysis of performance, we compare the score achieved through both methods. The participant's beats per minute (BPM) is taken three times each, prior to and after both studies to determine the heart rate, for a total of 12 readings per participant. Even though heart rate data can be further improved by coupling it with accelerometer, we determined that since the movement mechanic for both of the methods are fundamentally different, using an accelerometer is not suitable. Furthermore, BPM data can be easily obtained from various health monitoring devices, in our case, the Samsung Galaxy S6 Edge+ smart phone. Thus, data from both the NASA task load and heart rate monitoring are sufficient and relatively accurate to determine energy consumption [43, 54, 162]. Additionally, we use both the NASA task load and heart rate because relying on only one of them may not

Figure 4.6: User study for ArmSwingVR (top) and WIP locomotion (bottom).

achieve the accuracy we desire. Relying on a purely qualitative analysis is rather subjective, while heart rate monitoring is also associated with stress level [160]. However, for the purpose of this user study, since the participants will mostly be actively engaging in the VR environment using both locomotion methods, it is more than likely that the rise in BPM is due to physical activity. A total of 18 participants were recruited, comprising of 12 males and 6 females aged between 20 to 27. All of them do not have prior cardio-related health issues related and are inexperienced with both mechanics of using arm swing and WIP.

Results

Comparison was already made between actual walking and WIP [156], however, arm swing is a novel method that has not been developed or studies up previously when it comes to VR locomotion. Starting with the overall score for the participants, the average score for using arm swing is 7.44 while for WIP is 7.22, suggesting that they both perform equally under equal locomotion speed. Figure 4.7 shows the SSQ analysis results for both methods based on the SSQ computation method [69] with regards to nausea, oculomotor, disorientation, and the total score.

Interestingly, one of our assumptions was that arm swinging does not cause more motion sickness compared to VR-Step's solution. Yet, our results show that arm swing produces less sickness with regards to nausea and the total SSQ score, whereas WIP method shows to be better in terms of oculomotor by a small margin and disorientation. Disorientation in particular is worth mentioning, because the participants seems to be less disorientated compared to the rest state as well. A T-test analysis for each of the category between ArmSwingVR and WIP shows no significant difference except for nausea with a score of $p = 0.0022$. This was reinforced with an Analysis of Variance (ANOVA) for nausea that showed statistical significance between pre-test, ArmSwingVR and WIP ($F(2) = 10.951, p = 2.92x10^{-10}$).

In terms of presence, the results are divided into the following sub categories; realism, possibility to act, quality of interface, possibility to examine and self-evaluation of performance. Figure 4.8 shows the participants' feedback using the presence questionnaire for both methods.

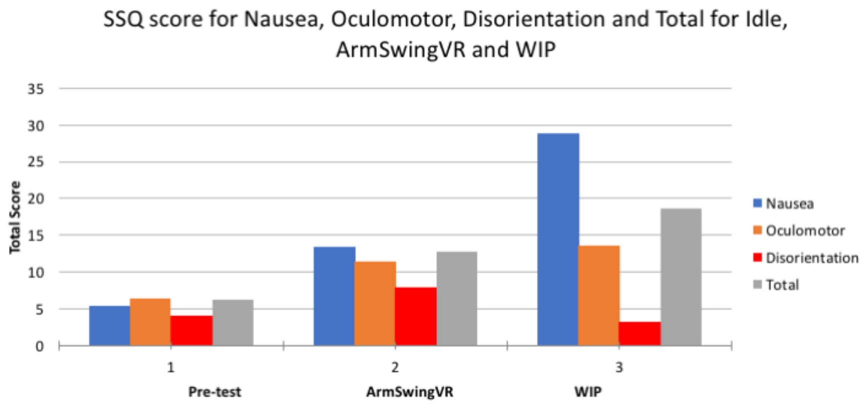

Figure 4.7: Chart for SSQ score.

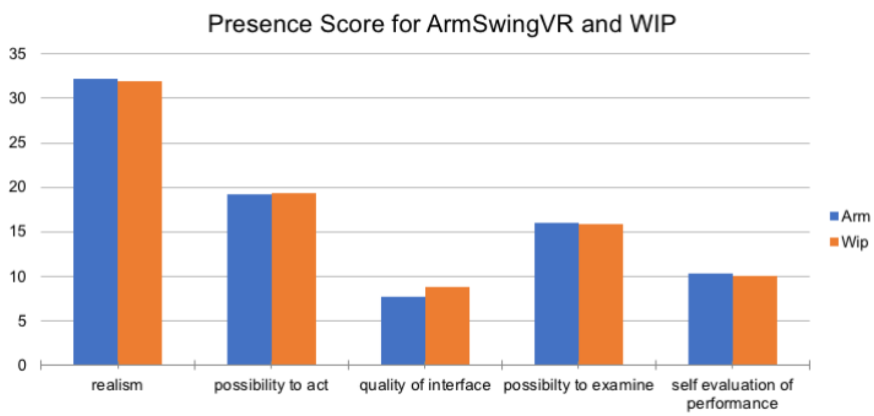

Figure 4.8: Chart for Presence score.

Performing a T-test for these forms of presence gives us a p score of 0.94, 0.97, 0.37, 0.88 and 0.81 respectively, proving that there exists no statistical significance between arm swinging and WIP. This also means that no significant sense of presence was sacrificed for the participants and the immersion level is comparable and preserved.

By observing the heart rate, we can clearly indicate that the WIP method causes a much higher BPM for the participants since it requires more motion and energy for a continuous jogging motion, as opposed to arm swing. Figure 4.9 shows the general BPM readings for the participants. Figure 4.10 shows the rise in heart rate for the participants for both arm swing and WIP, by subtracting the heartrate with its pre-test values. Overall, it can be seen that the highest rise in heart rate for a participant is 51.66, which was a rise from 78.67 BPM to 130.33 BPM for WIP approach. Arm swing has a significantly less rise in heart rate with the highest value being at 24.33, which was a rise from 87 BPM to 120.33 BPM. This clearly indicates that WIP methods generates a higher heart rate, which is associated to a higher expenditure of energy. To enforce this, the NASA task load data in figure 4.11 illustrates a quantitative take on this analysis on energy expenditure. This score is taken after each participant answers the questionnaire for mental demand, physical demand, temporal demand, performance, effort, and frustration, followed by weighting each of these factors.

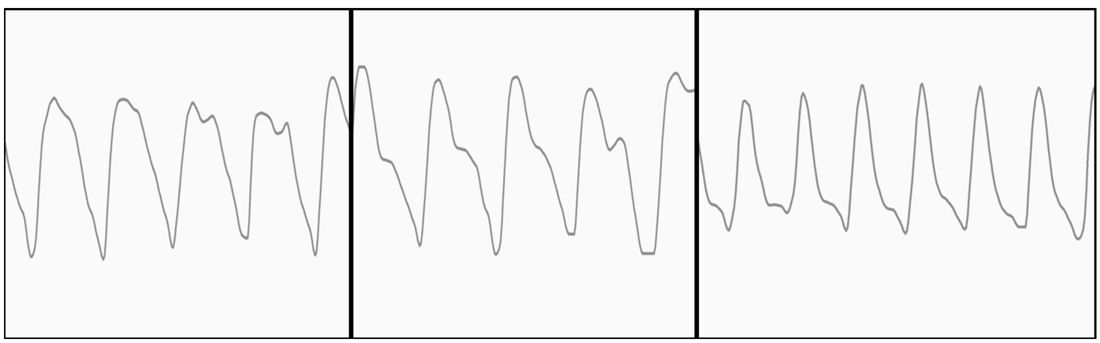

Figure 4.9: Heart rate against time during idle (left) around 80 BPM, ArmSwingVR (middle) around 95 BPM, and WIP (right) around 120 BPM.

It can be seen that the participants unanimously score higher for the WIP solution with the highest score at 65.67. The average ArmSwingVR score is 39.87, while for WIP is 54.18.

Discussion and Limitation

We showed the feasibility of arm swing to replace WIP. Studies have already proved that dynamic walking overall induces less motion sickness compared to static walking in virtual environments [63]. This can also be mentioned as a comparison between room scale VR or a wide AR space, with sitting VR where physical motion is kept to a minimum. According to the feedback of some of the participants, almost all of them agree that locomotion by arm swing is low in pro-

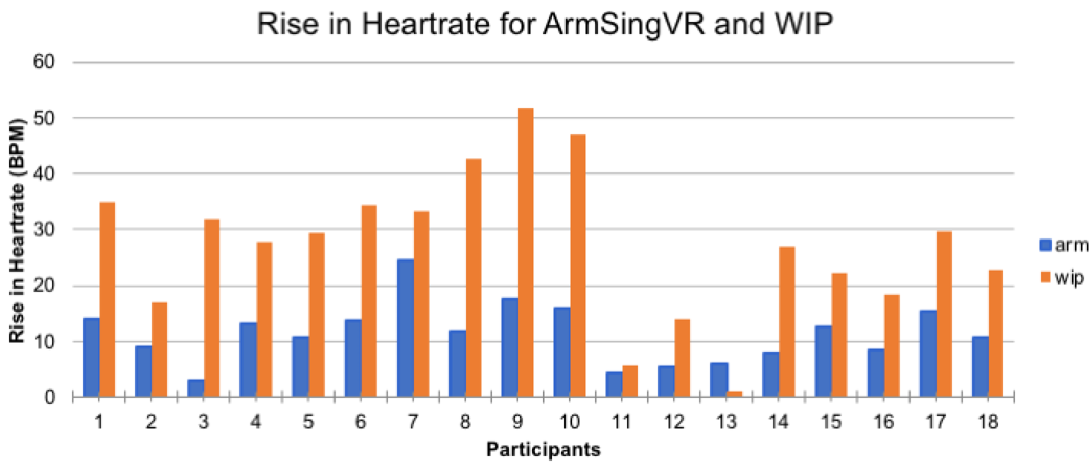

Figure 4.10: Chart for each participant's rise in heart rate.

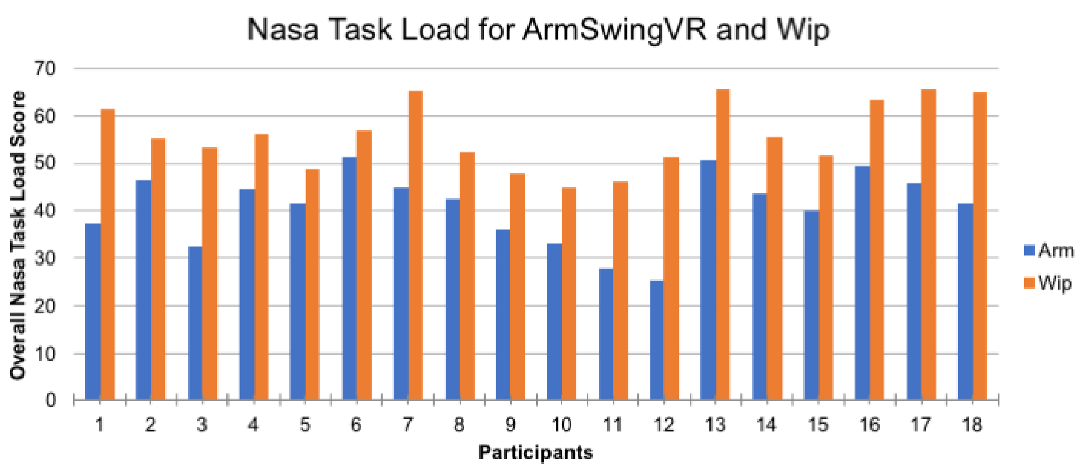

Figure 4.11: Chart for overall Nasa Task Load score.

file yet remains immersive, making it suitable for non-intrusive use. Furthermore, cables on HMDs, particularly in VR, caused some annoyance for WIP and rarely for arm swing, though as the technology evolves, wireless solutions are more than likely. Some of the participants noted that there is some gliding issue with regards to both methods. Gliding is when the participant stops moving, and the system takes an additional second to actually stop. This gliding issue does cause some motion sickness. However, similar to VR-Step, the gliding issue is not perceived once variable speed control is activated [149]. This is because even though gliding may still exist, since the system accelerates and decelerates according to the user, it is more difficult to notice. The immersion factor of WIP cannot be ignored, however, a continuous jogging motion is quite tiring as most if not all participants ended up sweaty and panting after just 15 minutes in the VR session. This is the reason why nausea scored high for the WIP method, since sweating was taken into consideration in its scoring. Another issue with WIP that is worth mentioning is that since the jogging motion uses feet movement (even though it is not obligatory for the system to function), most if not all participants tend to drift from their original position. This drifting often causes them to move away from the Vive's tracking area, or in some occasions, minor collision with the physical wall of the room. This is one of the factors that make WIP methods less desirable for social space usage.

For arm swing, most participants had no issue performing it for 15 minutes. Each participant was also clearly told to hold the controllers facing forward since the walking direction is influenced by it. However, some of them start to hold the controllers in other angles particularly after about 10 minutes, causing them to not walk straight, thus inducing a small but negligible amount of sickness. However, most participants also added that this feature is more realistic. After trying on the WIP method, they tried to do the same, and this caused some motion sickness since WIP methods rely solely on facing direction. Since arm swinging requires precise tracking of the hand trackers and HMD, it is vital for the participant to maintain in the Vive's tracking area. Thankfully, since no feet movement was present, the drifting issue can be avoided. Occlusion may still happen occasionally, but this is more of an issue for any infrared (IR)-based tracking. If sensing is based on inertial sensors, this issue is mitigated.

Overall, all participants agree that using arm swing is the better choice for VR navigation use even in social spaces, unless the VR environment was designed for working out or consuming energy. Furthermore, since some degree of physical interaction is required for locomotion as opposed to more traditional means like button input and blink teleport, the immersion is preserved. This allows for a wider target audience including the elderly to indulge in a more immersive VR experience.

Naturally, the main drawback of arm swinging at this point of time is that it is only limited to VR systems that provide controllers that tracks both hand positions. However, as AR and VR becomes progressively better and more main stream, such controllers will surely be adapted in other VR solutions, until the point where gesture-based recognition becomes mainstream. Additionally, arm swing was developed purely for navigation on a terrain, and was not designed for jumping because there is no noticeable or distinct arm motion when a human jumps. In this regard, WIP systems should be able to perform better, though this is another matter entirely but worth mentioning. Lastly, since the arms need to be constantly swinging, it is difficult to perform other forms of interactions that requires hand gestures while navigating. It is possible to use a single arm swing, though that is unnatural. Although the same can be said with most other WIP methods, it is nevertheless a matter that needs to be considered depending on the application. Even though sitting VR experience was not covered in this study, nevertheless it is worth considering. The algorithm currently used for our method does not support sitting experience at the moment, though that can be easily added on. Interestingly, some of the participants did mention that arm swing while sitting would be quite acceptable as opposed to head bobbing motion which is strange with no leg movement. This is best experienced with a chair that can rotate for turning around.

As mentioned previously, seeing how different AR and VR systems can be, it is difficult if not impossible to find one best locomotion method for all scenarios and applications. The proposed method manages to reduce energy consumption while still being immersive, making it suitable for wider audience of a different age gap, as well as for social spaces. Due to its software-based solution, users do not need to rely on additional peripherals and sensing methods like sensor-equipped shoes, etc. Finally, it is a convex interaction alternative to WIP that proves that even with the lack of the primary motion, users can adapt to this motion for locomotion. making it more efficient in terms of energy and space compared to conventional walking.

4.3.2 PinchMove

This section is adapted from a full paper that will be presented at the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2018). The paper was co-authored by myself, Zikun Chen, Liwei Chan, Megumi Isogai, Hideaki Kimata and Kai Kunze. PinchMove is another locomotion mechanic that, like Arm Swing, falls into the intimate interaction space. However, the mapping is changed here where the motions of the primary motion are mapped onto the arms, making the arms behave more like footsteps. Therefore, the motion type is not changed and the output remains linear.

This next work looks at another form of Convex Interaction utilizing the arms for locomotion. However, the difference is that, instead of prioritizing immersion, we look into the idea of accuracy of movement and aim to propose and accurate, convexed interaction mechanic. In today's virtual and augmented reality systems, it is nearly impossible to create a perfect locomotion mechanic suited for all types of applications and scenarios. Large virtual environments may benefit from more instantaneous and quick navigation, whereas small confined spaces may prefer to just use actual physical walking, depending on the hardware support. However, for spaces that are larger than that, yet still confined and requires precision accuracy, a suitable locomotion mechanic is still unknown. The currently preferred method for VR, teleportation, works well in mitigating simulator sickness and navigating the user in large virtual environments. However, it is not possible to perform accurate movements; which is acceptable since it was never designed to achieve that in the first place. On the other hand, physically walking is arguably still the best method, but requires a physical area as big as its virtual counterpart, equipped with precise trackers. Therefore, what is the best form of locomotion for a room-spaced environment, like an office, workshop, or lab, where the user requires accurate control of his or her position for professional use or simulation?

We propose PinchMove, a convex interaction mechanic that aims to address this by using a pinch gesture to grab a position in space and drag the user based on the change in position of the user's hand. This form of manipulation of the viewpoint is accurate and allows the user to move in four directions freely, as well as rotate depending on the angle of a pinch. It is based on a discrete style of locomotion as opposed to controlling the rate of movement (joystick controls). Furthermore, it is also suitable for users who are in a confined physical space, yet still, require precise control of his or her location in the virtual world since only arm movements are required.

Among the numerous available options for navigating in a virtual environment, most of them cater to either avoiding simulator sickness, being fast, being realistic, or all three at once. These are important factors to consider, yet at the end of the day, the best locomotion method is highly dependent on the application itself. Furthermore, an additional dimension that we wish to consider in this work is the precision in locomotion or the ability given to the user to precisely navigate to the desired point, given a moderately limited space.

Arguably the most popular locomotion mechanic for VR today is teleportation, which causes an instant change in position using a fade and blink animation [14]. A variation to this is Dash, where instead of teleporting, the user moves at high speed to the designated target location to provide a better sense of movement. This variation in speed and transition for teleportation could mean that a proper tweak in speed and transition can improve the experience (effects of speed and transition on target-based travel techniques). Interestingly, it was still found that regular teleportation yielded the least discomfort and that transition techniques like animations actually do not significantly effect performance or cyber sickness. One of the currently available solutions most related to our proposed method is the grabbing locomotion, where the user grabs the space in front of them to traverse. This can be often seen in climbing VR games like Climbey and The Climb, which allows the user to traverse vertically. The factors to consider in an implemented locomotion mechanic usually boils down to simulator sickness, quickness, and sense of realism. However, one other factor less mentioned is the consideration for precision, or how much degree of control a user can be given to reposition himself or herself as accurately as possible.

Simulator sickness, cybersickness or motion sickness, are terms used particularly in VR environments to mean nausea or dizziness, often caused by the 'sensory conflict theory' that refers to how our vision is augmented to receive motion signals, yet our non-vestibular proprioceptive senses don't, resulting in a variance that causes said sickness ([49, 134]. To help curb this issue, there has been an established guideline for VR development, such as how movement acceleration should be linear and not too long, avoid rotation on the forward-axis, avoid yaw-axis rotation, and avoid direct control of the main camera view [95]. Among the current solutions are adding a motion platform, performing direct vestibular stimulation, and implementing rest or static frames, of which only the last option can be considered to avoid any additional peripherals [78]. Furthermore, over the years, VR researchers have begun implementing several visual tricks, one of them being 'grounding', where part of the foreground remains static [7]. Essentially, it reduces the immersion level of the user through methods like reducing the field-of-view (FOV) [42], or adding a static visual frame for the user. This can greatly reduce the effects of visual update delays and simulator sickness compared to a wider FOV [35]. Finding an optimum FOV can be tricky because reducing it sacrifices immersion and requires the user to perform a bigger head and eye movement to view content [159]. However, it was also mentioned that a smaller FOV is more acceptable for smaller virtual environment compared to larger ones. Fernandes et al. found that a FOV of 80°was the limit of an acceptable FOV before it started to detract the experience [42]. Among some of the other methods explored by researchers are using an independent visual background (IVB) which is similar to Google Daydream's implementation of a virtual meta world, though it was only proven to work in driving simulations at this point of time [38].

The idea of using a pinch based gesture interface existed since the first creation of the glove input device, dating back to 1962. Then, IBM was rewarded a patent for such a glove that has sensors on each finger, allowing for up to 1, 048, 575 possible input combinations [125]. Since then, various other glove-based devices like the Z-Glove and DataGlove [174] that uses ultrasonic positioning and magnetic positioning respectively were developed to achieve finger-based gesture recognition. Even though these devices allow whole hand interaction [145], pinching between 2 fingers is the most minimal form of performing a gesture that provides proper haptic feedback, when compared to other gestures like grasping or grabbing, or bimanual gestures like clapping. Even though it is most commonly seen in touch screen interfaces for zooming, it has also been explored for applications like navigation. A two-handed navigation system was developed using the PinchGlove where the relative vectors between the hands allows the user to to determine the direction and pinching allows navigation. A different pinch gesture allows velocity control. Even though the technique was deemed flexible, no proper studies were conducted and since it controls the rate of movement as opposed to discrete positioning, it suffered from accuracy. Relying on various pinch gestures using multiple fingers may provide additional functions, yet at the same time make it less intuitive if the user needs to memorize each function. FingARtips focuses on the pinch between index and thumb for grabbing, pointing, and pressing in an AR environment [15]. Even though there was no issue with the pinch gesture, the system was limited instead by the tracking of the AR system used. Another previous work used pinch gestures to manipulate computer aided design (CAD) models using bimanual pinch gestures. GaFinC combined gaze with pinch gestures for selection and manipulation, with different pinch movements resulting in translation and rotation of the 3D model. However, particularly for rotation, the user must use two hands, as one handed operation is only limited to translational operation.

Implementation

Our system was implemented using the Oculus Rift CV1 due to how the Oculus Touch controllers conform naturally around the users' hands to perform pinch based gestures. As of this moment, only positionally tracked controllers can be used, though we believe that future iterations of AR and VR input will provide this input method for all types of VR experiences. The user simply needs to rest the thumb on any of the face buttons or the analogue stick, while pressing the index trigger. This creates a pinch animation on the virtual hands as well. We divided the implementation based on two methods; one-handed and bimanual, depending on the user's preference. The pinch gesture works by "pinching" the viewport or any point in space and dragging it towards yourself, creating an inverted movement that causes the user to navigate to the direction the controller was previously. A similar example would be grabbing a rope and pulling it while sitting on a wheeled-office chair. Though not the most popular navigation method for VR, a variation of this method can be seen in VR games like Climbey [81] and The Climb [31], because it emulates the arm movements of actual wall climbing. Grabbing a point and lifting up moves the user upwards, akin to PinchMove. However, the aforementioned games differentiate in two factors; they allow only vertical movement, and rotation is not possible. The main benefit for this particular form of navigation is that since the user moves depending on the position of the pinch in world space, he or she can perform accurate positioning through arm movement alone. For our implementation, we choose to disable vertical movement, or movement about the y-axis since we are focusing on ground-based navigation for this study, though it can easily be added depending on application.

To allow the user to move forward by pinching the space, we first find the initial position of the controller the frame the trigger is pressed, and final position after the controller moves to a new position. This difference in Euclidean space is added into the user's current position, allowing them to move front, back, left and right depending on the controller's movement vector. Once the user has decided on the final position and releases the trigger, movement is halted and their final position becomes the new current position. therefore, if a user places their hand on a virtual object and pinches themselves towards it, they will be standing directly in front of the object based on the desired distance.

One-Handed navigation ensures that the user can fully control his or her position in virtual space using a single Touch controller. This leaves the second hand to be free for other tasks. However, care needs to be taken when mapping functions for translation and rotation to avoid unintended navigation. The index finger trigger allows the user to perform pinch translation, whereas the middle finger trigger allows the user to rotate about the y-axis. The center of rotation is assigned to the controller instead of the user, where the user actually orbits around the controller during rotation. Orbital-like movements in VR has been explored before and has been used as a primary navigation mechanic before [98]. However, the applied orbital movement mechanic is more subtle since the controller is never too far away from the user. The angle of rotation is based on the controller orientation about the y-axis. When the controller is rotated around the user, it is as though the viewport rotates along with it, though in actuality the user has successfully rotated about the controller. Therefore, if the user performs this movement when placing the controller on a virtual object facing another direction, they will be able to reposition themselves in front of it at any desired orientation. If the center of rotation was placed at the user's position as opposed to the controller position, this is more conventional but accurate positioning would not be possible.

Bimanual navigation allows the user to navigate either with the left controller, right controller, or both at the same time. Either left or right translation would behave no different from one-handed navigation. However, for bimanual, we take the average position of both controllers as reference for movement. If one controller travels further than the other, then rotation is triggered, where the user rotates about the reference object. Therefore, if the user performs this movement when placing both controllers on a virtual object, they will be able to reposition themselves accurately based on the movements of both of their hands. Naturally, enabling PinchMove for both hands removes the ability of the user to use a free hand for other task. However, we wish to evaluate the performance comparison of these two methods and determine the users' preference with regards to accuracy.

To ensure that simulator sickness is minimized, we employ the tunneling method for PinchMove. Tunneling is enabled through the vignetting and chromatic aberration visual effect attached to the main camera in the scene. A vignetting intensity of 0 creates a maximum FOV for the user, in this case 110°which is the FOV of the Oculus CV1 headset. When the vignetting intensity is increased to 1, the screen is completely blacked out, equaling to an FOV of 0°. The two important tunneling parameters are therefore the intensity as well as the rate of which the tunneling effect appears, as discussed by Fernandes et, al [42]. In this work, it was determined that an 80°FOV was the minimum allowable FOV before it starts to detract from the experience. Even though the rate of tunneling was also determined, movement in PinchMove is largely different then regular controller input where the rate of movement is controlled, whereas PinchMove is more discrete. At the beginning of the tunneling effect, it reaches the highest speed initially, then slowly decelerates as it approaches 80°FOV. Therefore, a pilot study was designed to determine the appropriate maximum speed for tunneling, and will be further explained in the following section.

User Study

The user study is divided into a pilot study that focuses on determining the right parameter for tunneling to minimize motion sickness, and the main study

Figure 4.12: Environment for the pilot study. The left image is the top view with all waypoints position shown, whereas the right image shows the participant's view during the study.

that focuses on evaluating the accuracy of positioning and orientation. We chose to use the Oculus Rift CV1 with the Oculus Touch controllers due to its more ergonomic and natural controller shape for pinching, though our implementation has been tested with the HTC Vive as well.

For this study, the goal is to determine if the proposed navigation method causes a significant amount of simulator sickness depending on the speed of the tunneling effect. Although there exist several methods to mitigate sickness in VR, we chose the conventional tunneling effect of limiting the FOV of the user as it has been proven time and again to be effective [42]. Fernandes et, al. found that in a pilot study, a FOV of 90°is deemed to be the preferred minimum FOV whereas 80°is the largest FOV to detract from the experience. Therefore, we choose to use an FOV of 80°so that it would not further detract the experience. Furthermore, a subtle and slow decrease in the FOV restrictor even causes some of the participants to not notice the change in FOV. However, a major difference between that study and ours is that the study was designed for the participants to navigate a huge virtual space, whereas PinchMove was designed for a more confined area. Secondly, pinching the viewport results in a more discrete style of navigation as opposed to regular navigation that is continuous and works by controlling the movement rate. The tunneling effect tends to only appear when the user is moving in the environment, and slowly dissipates when the user is static. Therefore, the tunneling effect would appear at a slower rate as opposed to conventional method, since releasing the pinch gesture dissipates the tunneling.

We recruited a total of 8 participants for the pilot study. To determine the appropriate tunneling rate and sickness, we designed an office-like environment where the user is required to navigate using PinchMove for two minutes. We also predefined three levels of tunneling speed, slow (maximum of 10.15 degree per second), medium (maximum 15.18 degree per second) and fast (maximum of 30.08 degree per second) by altering the tunneling smoothing time from 110°to 80°. At the beginning and end of each session, each participant is required to answer the Simulator Sickness Questionnaire (SSQ) to estimate the current nausea, oculomotor, disorientation, total sickness score, and total rise in sickness that is being felt [69]. During the study, the participant is required to navigate through white capsule waypoints that appears one at a time in a confined office space which are placed in a way that forces them to translate and rotate the viewpoint, similar to the experiment by Fernandes et, al. as shown in Fig 4.12. At the end of each session, each participant is also required to answer another set of questionnaires designed by Suma et, al. to determine if the change in FOV was noticeable or not [146]. Participants were asked to rate the following questions from a scale of 1 to 7, where 1 means "I did not notice anything" and 7 means "I obviously noticed it."

1. I saw the virtual environment get smaller or larger.
2. I saw the virtual environment flicker.
3. I saw the virtual environment get brighter or dimmer.
4. I saw that something in the virtual environment had changed size.
5. I felt like my field of view was changing in size.
6. I felt like I was getting bigger or smaller.
7. I saw that something in the virtual environment had changed size.

To determine the accuracy of navigation, we designed a virtual environment for the participant to navigate using PinchMove. To determine the accuracy of movement, each participant is required to complete a task where the time required to complete each trial is recorded. The task required the user to align a user object with a target object. The user object is a cube placed in front of the user that

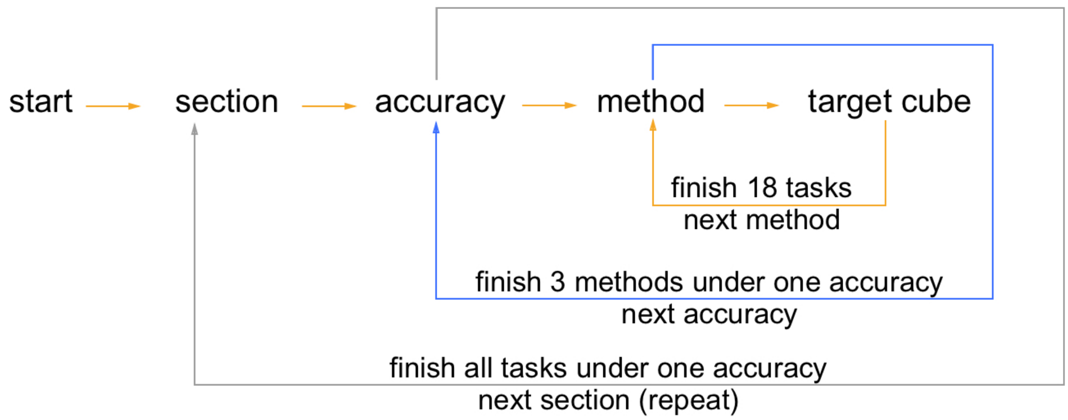

Figure 4.13: Flow Chart for the designed user study

follows the user's translation and rotation (blue cube). The target cube is a cube placed in the virtual environment at a predefined position and orientation (semi-transparent cube with an orange face to depict the front of the cube), illustrated in Fig 4.15. To determine the possible predefined locations, we consider the FOV (3 angles) and distance from the user (2 distances) to determine 6 possible positions. We use FOV to ensure that the target cube is always visible to the participant at the beginning of each trial, and we select distances according to the standard arm length for near-field interactions. There is a possibility of 3 orientations per position (-45°, 0°, and 45°). We also fix three accuracy levels for the user for each trial (98%, 96% and 94%) for the position and orientation, which can be seen on Fig 4.14. For the position accuracy, an accuracy of 100% means that the center point of the user object is exactly at the center point of the target object, or when $d = 0$. 0% accuracy is the furthest possible distance between the two center points when they are in in contact, which is also equivalent to the distance between the center point and the middle point of the edge of the target cube (when $d > s$). For the orientation accuracy, an accuracy of 100% means that the angle between the user object and target object, α, is 0°. 0% accuracy is the largest possible angle deviation between the two objects. Since we measure the acute angle as the difference in orientation, 90°is deemed 0% for orientation accuracy (when $\alpha \geqslant 90$). Finally, each trial is repeated three times. The three input methods are described below:

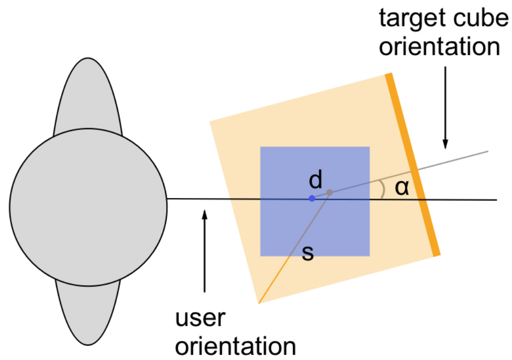

Figure 4.14: Parameters in defining the accuracy of translation and rotation

- GamePad: A conventional locomotion mechanic utilizing the thumbsticks on the gamepad to move and rotate. This input method serves as the standard baseline of comparison.

- One-Handed: In addition to navigation with the index trigger, the middle finger trigger is used to enable rotation. The participant will be seated on a leg chair for this scenario.

- Bimanual: Both controllers are required to rotate the participant by the relative angle between them, without any additional button. The participant will be seated on a leg chair for this scenario.

Only one target cube will appear at a given time, and the next cube will only appear once the participant successfully completes that trial. The total amount of trials per participant is 6(positions) x 3(orientation) x 3(inputs) x 3(accuracy) = 162 trials per participant. In this study, the independent variables are the input methods, accuracy, position and orientation of the cube. The dependent variable is the time required to complete the trial. Fig 4.16 shows the view of the participant during the user study.

Prior to the study, the participants are allowed to familiarize themselves with the navigation mechanic for 5 minutes. During this period, we also perform a quick calibration for each user to collect the speed of movement during PinchMove. We then modify the gamepad input speed to be equal to the average speed of PinchMove for each participant. We recruited a total of 19 participants (10 male,

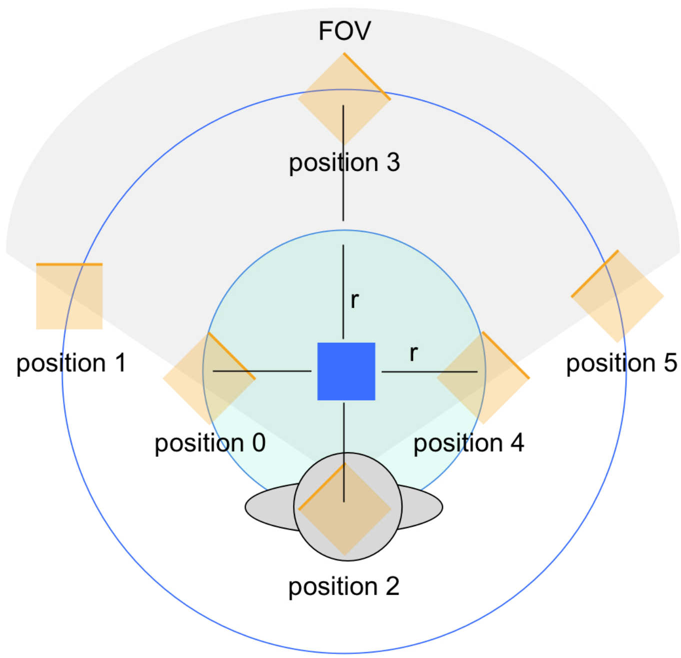

Figure 4.15: The positions of the target cube (only 1 can be seen at a given time) for the user study with 3 possible orientations.

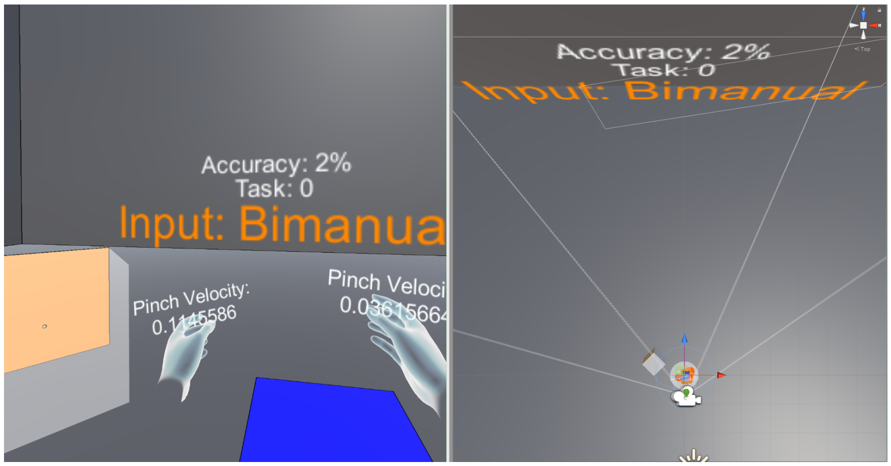

Figure 4.16: The participant's view of the study and a top-down view of the environment. The target cube can be seen at Position 0

9 female) aged between 22 to 34 years (mean: 25.63, SD: 2.81). At the end of the study, each participant finally answers a Likert-scale questionnaire from a rating of 1 to 5 for perceived accuracy, efficiency, reliability, learnability and likability for each navigation method.

Results

For the pilot study, Fig 4.17 show that a medium tunneling speed overall caused the least rise in motion sickness with a score of 10.285 compared to a higher tunneling speed with a rising score of 17.3 and a slower tunneling speed with a rising score of 24.78. Looking at the average score for nausea, oculomotor, disorientation and overall total score, the medium tunneling speed scores overall causes less sickness as well. The data were analyzed using repeated-measures ANOVA. The repeated measures ANOVA reveals that there is no difference found on increase score between the treatments ($F_{2,14} = 1.57$; p = 0.243). Interestingly, a faster tunneling speed scored lower than slow tunneling speed in all these categories except for the average increase of total sickness. For the noticibility questionnaire, the only relevant questions were the third and fifth questions which are related to the change in FOV. According to Fig 4.18, participants were much quicker to realize that the environment seemed darker at the fastest tunneling speed whereas the slow and medium speed were about equally slow to realize. However, most

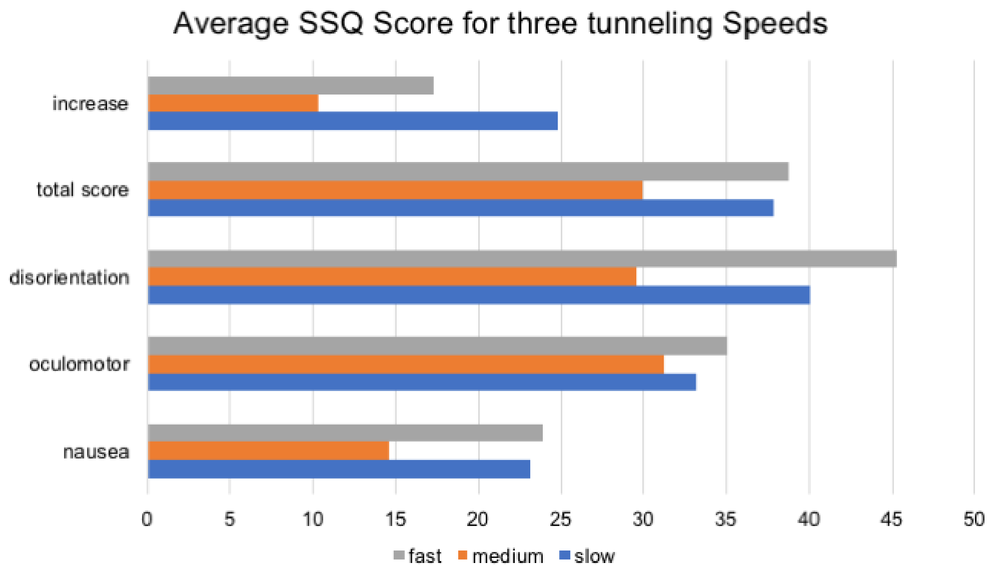

Figure 4.17: SSQ results

participants realize the change in FOV at the medium speed, followed by fast, and finally slow.

For the main user study, the results were analyzed using ANOVA repeated measures where we evaluated the accuracy of each input method based on completion time of the predefined accuracy. All post-hoc comparisons used Bonferroni corrected confidence intervals. There was a significant main effect of accuracy ($F_{2,36}$ = 7.250; p = 0.002 < 0.005) and pairwise tests show that users were significantly faster with an accuracy of 94% compared to 98% (6.326 vs. 8.329, p = 0.002 < 0.005). There was also a significant main effect of interface ($F_{2,36}$ = 3.41; p = 0.002 < 0.005). Pairwise test found that bimanual input was faster than one-handed input (6.849 vs. 8.183, p = 0.004 < 0.005). However, there was no significant difference between gamepad and bimanual, as well as gamepad with one-handed.

For the final qualitative questionnaire, Fig 4.20 shows that bimanual was at average, generally perceived to be the most accurate, efficient, and likable. Gamepad input was slightly, but not significantly, higher than bimanual in terms of reliability, but is noticeably higher in terms of learnability. For perceived accuracy, bimanual was slightly, though not significantly higher than gamepad, with the

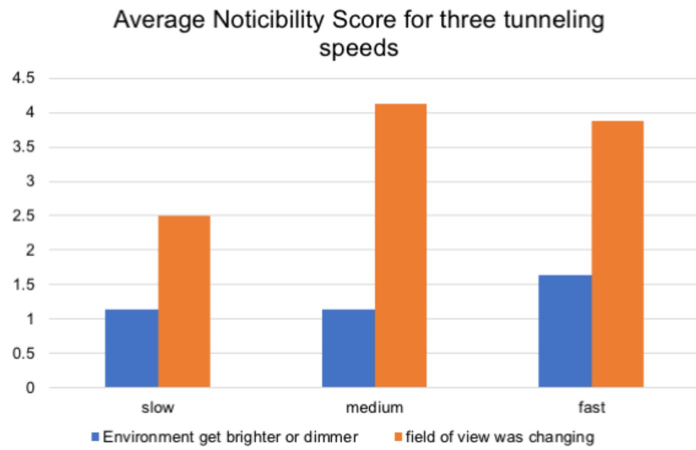

Figure 4.18: Noticibility Score

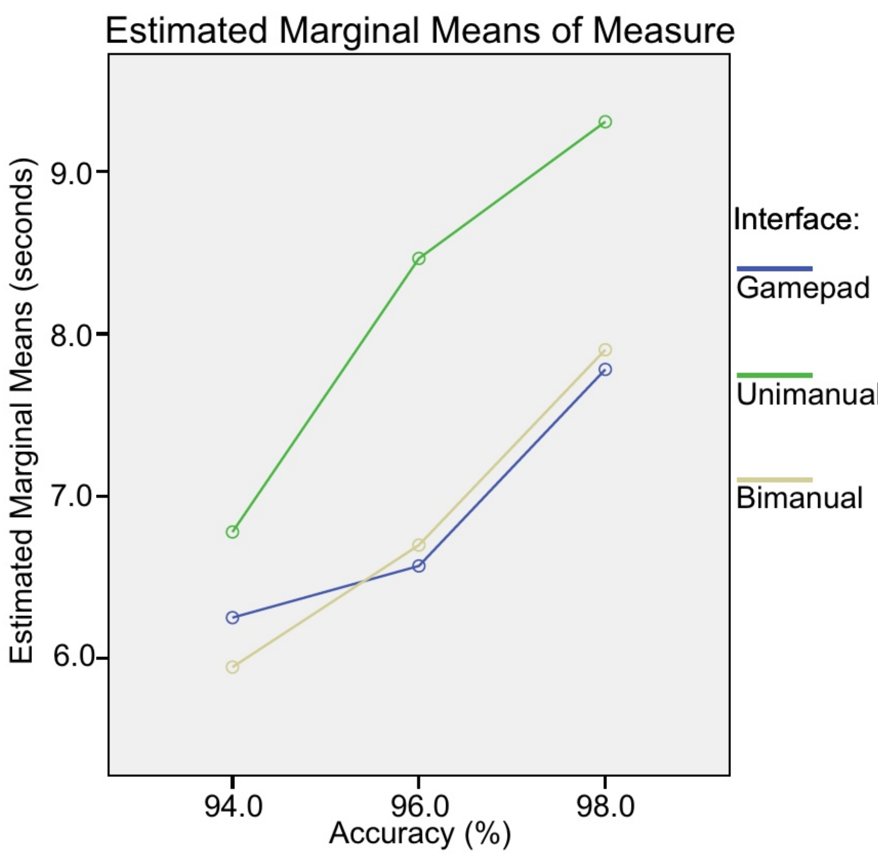

Figure 4.19: Estimated Marginal Mean of three input methods according to accuracy

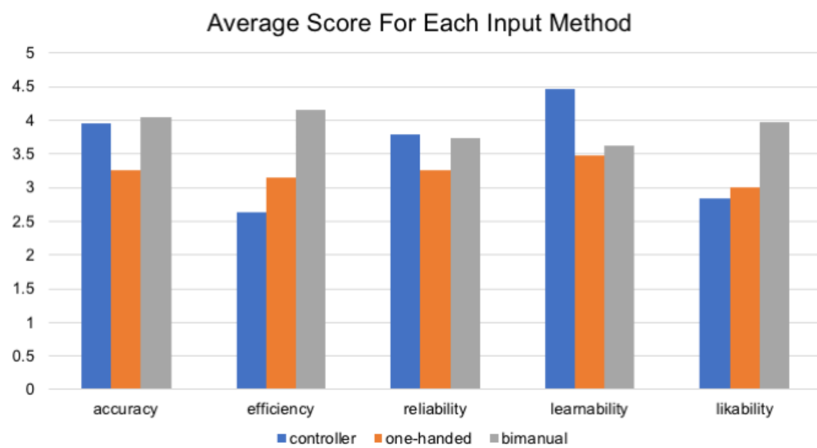

Figure 4.20: Qualitative results

one-handed method being the least accurate. Significant differences were found in accuracy, efficiency, likability, and learnablity, but not in reliability. For accuracy ($F_{2,36} = 5.16$; $p < 0.05$), significance exist between gamepad and one-handed ($p < 0.05$) as well as one-handed and bimanual ($p < 0.05$). For efficiency ($F_{2,36} = 9.38$; $p < 0.001$), significance can be seen between gamepad and bimanual ($p < 0.05$), and between one-handed and bimanual ($p < 0.05$). For likability ($F_{2,36} = 4.0$; $p < 0.05$), there is a significant difference between one-handed and bimanual ($p < 0.05$). Finally, for learnability ($F_{2,36} = 7.317$; $p < 0.005$), there is a significant difference between gamepad and bimanual ($p < 0.05$), and between one-handed and bimanual ($p < 0.05$). There is no significant difference for reliability ($F_{2,36} = 2.520$; $p = 0.095$).

Discussion and Limitation

Overall, we showed the feasibility of PinchMove as an accurate navigation mechanic for virtual environments. For the main study, we asked the participants their general experience in VR, informally rank their preferred input methods from best to worst, overall comment on the user study, as well as suggestions for scenarios that can benefit PinchMove. Out of the 19 participants, 14 preferred PinchMove over conventional gamepad, and 12 from them preferred bimanual input over one-handed. The 5 participants who preferred gamepad mainly said that it was simply due to it being more conventional and common, leading to lesser time to master since the operation does not need to be understood. PinchMove also lead to fatigue of the arm since each user was required to keep up with the pinch and pull motion for nearly an hour, though this is only due to the period of the user study and does not reflect an actual use-case scenario. Finally, another common feedback is that since PinchMove relies on pulling as opposed to gamepad which is more akin to pushing, the inverted controls presented some difficulties for these participants, similar to scrolling on a touchscreen where scrolling upwards with the finger causes the screen to scroll downwards instead.

The users who prefer PinchMove navigation presented some interesting related scenarios. One participant mentioned that, a gamepad was similar to sitting in a car and moving forwards, whereas PinchMove is more like grabbing the road and pulling it to move forward. Furthermore, PinchMove allows fine control of movement speed, where the user moves as fast as their hand, as opposed to gamepad with a predefined maximum speed. Compared to the conventional gamepad, PinchMove was overall more enjoyable to use.

Bimanual was overall preferred because it provided a better spatial sense during movement, and using both hands for positioning overall feels easier and more intuitive. A common negative feedback for bimanual navigation is that since only the index trigger was used, participants who moved too fast caused accidental rotation, since they press the second controller trigger before releasing the first one for a split second. However, some participants also mentioned that since only a single trigger button was used for both controllers, no memorization was needed and it was easier to master. One of the participants who was experienced in VR, even used strategies for bimanual navigation for maximum speed, such as using a circular, pedaling motion between hands for quick movement, and orbiting one controller around another for quick rotation.

The 2 participants who preferred one-handed navigation agreed that even though it was more difficult to master compared to bimanual, it was overall more intuitive after the learning phase. However, it heavily depends on the scenario as well, since the one-hand option was provided so that the user is free to use another hand for other use anyway. They also mentioned that since translation and rotation was mapped to different buttons, no unintended movement can occur.

Interestingly, for PinchMove, participants overall prefer bimanual over one-handed because they claim that one-handed rotation felt inverted, even though rotation for both methods actually rotate at the same direction.

In terms of motion sickness, more participants claim to have felt more motion sick using gamepad, followed by one-hand, and finally bimanual. A participant claimed that pulling navigation overall presented less motion sickness compared to pushing navigation. It is a common design rule in virtual environment manipulation to avoid rotation about the upward, or y-axis as this can cause heavy sickness. Even though all input methods implement this, sickness was less for PinchMove since the rotation provided the participants with a sensation that they were rotating the world around them instead. Because of this, a participant even claimed that mastering PinchMove in return caused him or her to be more motion sick when using conventional gamepad. Finally, do understand Convex Interaction further, PinchMove changes the mapping type compared to ArmSwing, yet proved that users find this motion extremely intuitive. The efficiency of this method is derived from both space and time as shown in the results.

4.4. Convexed Locomotion: Head-Based Gestures

We go further down the space constrain route by looking into motions within the peripersonal space. AnyOrbit and GazeSphere, a variation of AnyOrbit combined with eye tracking, will be discussed in this section.

4.4.1 AnyOrbit

This subsection is adapted from a poster paper presented at the ACM Symposium on Spatial User Interaction 2016 (SUI 2016) [98]. The paper was co-authored by Benjamin Outram, myself, Kevin Fan, Kouta Minamizawa and Kai Kunze. This work was also further refined and accepted for publication at the ACM Symposium on Eye Tracking Research and Applications (ETRA 2018) which was co-authored by Benjamin Outram, myself, Tanner Person, Kouta Minamizawa and Kai Kunze. We now delve into using simply the head movement for Convex Interaction. Besides the extreme limitation of space for the motion, we also use a different motion type, which is neck rotation, to create a locomotion mechanic that is completely unconventional and non-linear, which is a orbital-type movement.

We propose AnyOrbit, an input and interaction mechanic using purely head rotation. Head mounted displays (HMDs) are being increasingly used to consume media such as games, film and sports, and advances in filming and computing technology are able to supply increasingly detailed data, including full 3D visualisation, of sports and other media [141]. There have been many approaches to navigating 3D visualisations, but in the context of immersive virtual environments (VEs) experienced through motion-tracked displays including HMDs, limitations in real space, tracking area, and the existence of simulator sickness, call for special considerations, and strategies remain limited. Since we are using purely head motion as an input mechanic, simulator sickness becomes a main concern. We demonstrate that simulator sickness can be mitigated by coupling head-rotation angle with lateral movement, compared to movement using a joystick. Such movement is consistent with movement along circular or spiral orbital arcs as the user rotates their head, with the speed proportional to the orbit radius. We demonstrate how such movement can be exploited for observation and navigation tasks in VEs experienced through HMDs, providing a versatile 3D navigation and observation modality, especially when controlled using real-time eye-tracking data.

Simulator sickness may not be as prominent in AR, but is a major concern for the adoption of VR and related technologies [93]. It has been the topic of significant research interest over the last several decades and is characterised by feelings of nausea caused in part by a mismatch between visual and vestibular stimuli [121]. Linear motion, acceleration and rotational motion in VEs all cause significant simulator sickness, presenting a challenge for creating navigation strategies in VEs [21]. To combat simulator sickness, a wide variety of techniques have been adopted. Teleportation and fixed-angle rotations reduce sickness associated with vection (apparent motion caused by visual stimuli). Another approach has been to use proprioceptive cues, which are considered important to reduce sickness and increase immersion and presence [111,118,144]. For example, the Walking-in-Place (WIP) technique has been shown to result in less sickness than movement with a joystick [64,150].

There is a precedence for linking head-rotation in particular to movement in VEs, not only because the proprioceptive cues reduce simulator sickness, but also to provide intuitive and hands-free interaction. The use of head rotation to directly control 6 degrees of freedom (6DOF) of linear motion and rotation was shown to produce less sickness and increases navigational task performance compared to joystick input [22]. Using head rotation to control orbital motion improved performance and user experience in a 3D object manipulation task in a desktop environment [61].

Orbital motion in particular, which is ubiquitous in CAD software systems, is instantly intuitive and particularly suited to observational tasks [71,97]. As Koller et, al. point out, perspective selection around an object can be achieved much faster and with less effort than conventional 'flying' metaphors, while maintaining the point of interest (POI) in sight at all times [75]. Head rotation was the preferred method out of several alternatives in a movement and observation task related to radiation therapy [25]. As well as CAD and data visualization, orbit-like motion is also represented by fly-by camera shots in film and sports coverage and in strafe-and-shoot strategies in first-person-shooter (FPS) computer games. This suggests that orbital-mode HMD techniques could be leveraged for these types of media experiences, for example, Mine suggested its use for a meta-CAD system [87]. There is a tendency towards 3D and free-viewpoint video formats and technologies [141], and many of the features of orbital viewing in HMDs make it ideal for interacting with such 3D formats compared to traditional devices such as flat-panel displays. The technique naturally makes the velocity of the user proportional to the distance from points-of-interest at the center of orbital motion, which is also advantageous for user control [83].

We therefore propose that the use of orbital motion controlled by head-rotation can be exploited to provide a navigation strategy with several key advantages:1) It is an effective convex interaction mechanic by being confined in the intimate proxemics, 2) It allows continuous movement while mitigating simulator sickness and 3) it is suitable to many use-cases in which orbital and sideways type motion is already employed.

Our approach was to build upon previous work on orbital navigation to make the technique more versatile as a navigation strategy. AnyOrbit exploits toroidal geometries to allow ideal spiral orbital paths to new POIs, allowing 6DOF navigation. Then, based on evidence that head-coupled movement reduces simulator sickness [22], we explored this deeper with a quantitative user study for orbit-like motion. Finally we explored a variety of use cases, finding that the use of eye-tracking for controlling the center of rotation provided a powerful technique of navigation and observation, and further finding that AnyOrbit can be used to guide users around environments and POIs without limiting their rotational freedom.

Implementation

Orbital techniques generally provide 3DOF of movement, two orbital directions and one radial direction, which limits the user's view to only the radial direction in towards the center. Orbital techniques are often then combined with flying, panning and other 'modes' that can be switched between [44, 147, 172] to allow other types of movement and 6DOF navigation. These methods work well for desktop CAD applications, but within an immersive VE context, panning and other types of motion as we have seen cause excessive motion sickness. To overcome this problem, Koller ,et al. suggests allowing the user to switch between traditional ego-centered rotation, and orbital viewing modes [75]. While in the ego-centered view, the user can select an object of interest to become the orbital center using a peripheral input method such as a control stick. The user is then forcibly moved to a location determined by their current rotation angle, the current radius between the user and the object, and the new selected center of orbit, or alternatively the user is locked into an orbital rotation such that their forward-facing vector is nonparallel to the orbital radial vector. A similar technique of switching between ego-centered and orbital rotation called 'Torbit' has also been adopted by Penrose Studio's short VR film entitled 'The Rose and I', and they suggest the technique could be developed for story-telling (Eugene Chung, Penrose Studios, December 2015).

AnyOrbit [98], which we will go into detail in this section, tackles the problem differently. The radius of the user's movement is controlled such that they move on a spiral path towards a new circular orbit about any chosen orbital center and radius, no matter in which direction they turn their head. Thus, the motion has several key desirable qualities motivated from our previous discussion and simula-

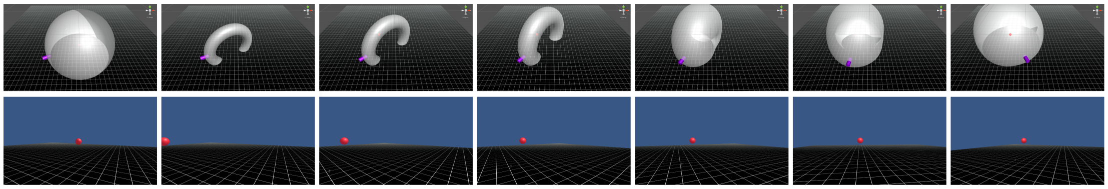

Figure 4.21: Illustrates how the calculated toroidal orbital surface changes as the user in the 3D environment moves to a new orbital center (the user's position as viewed from above is shown in figure 4.23 (a)). From left to right: The user (purple) is in a spherical orbit about the red marker in the center of their view. The marker is then moved to a new location in the left of the user's FOV. As the user rotates towards the marker, they move along the small radii of a toroidal surface. As the marker comes back to the center of the FOV, the user is now on a new spherical orbital surface about the new marker position. The dynamically controlled toroidal surface allows different radii of curvature in horizontal and vertical directions, depending on the position of the marker in the FOV, and produces smooth movement from any orbit to any other.

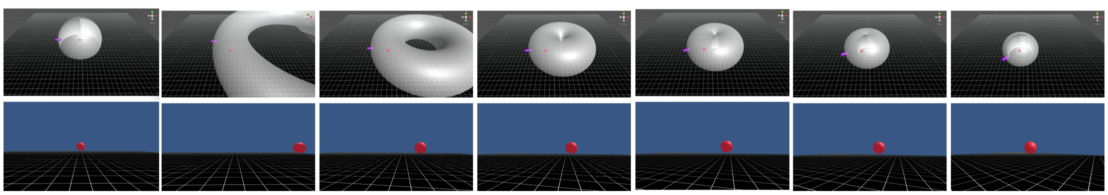

Figure 4.22: From left to right: As in figure 4.21, however in this case, the user rotates away from the marker, and the horizontal radius of the toroidal surface increases. As they rotate they follow a large orbit such that the marker returns to the center of their field of view, and they find themselves on a new spherical orbital surface about the new marker position. The user's corresponding position as viewed from above is shown in figure 4.23 (b))

tor sickness experiment: The motion is continuous, directly coupled to the user's head rotation, and always lateral to the users head. The result is that the user can smoothly and intuitively transition between any position and orientation on an ideal path, while continuously looking at the point of interest (POI) for which their destination perspective is being selected, all while experiencing less simulator sickness. The details of the algorithm and user experience will be described in the next section.

As we have discussed, a fixed orbital center, and only 3DOF of movement (radial, azimuthal and zenithal), precludes the user from navigating to any arbitrary 6DOF position and orientation. In AnyOrbit, we control the position of

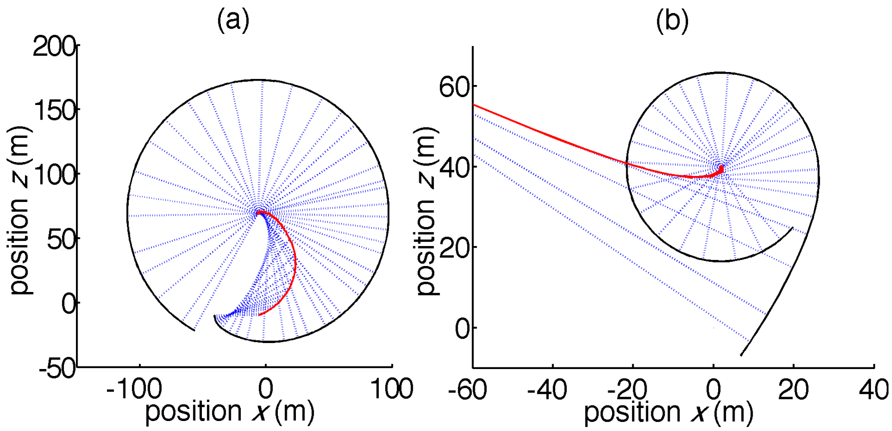

Figure 4.23: Illustrates the smooth path taken from one position and orientation to a position and orientation about a new orbital center. (a) and (b) show cases in which the user rotates towards and away from the new chosen orbital center location respectively. The black line indicates the path taken by the user, the red line indicates the path taken by the dynamically controlled orbital center about which they are rotating, and the dotted blue lines indicate the forward-facing vector of the user, which is alway perpendicular to the velocity and connects the users position to the orbital center. By controlling the orbital center dynamically, the users path spirals towards the new orbital path.

a **marker** that identifies the desired orbital center, (such a marker is important for user comprehension [44]). A problem arises that if the marker defines the center of rotation then the user would end up no longer moving perpendicular to their motion, since the marker is not in general directly forward and in the center of the user's field-of-view (FOV). Therefore, we shorten or lengthen the radius of curvature of the orbit depending on the location of the desired orbital center marker in the current FOV, and whether the user is rotating their head towards or away from the marker. In the case that the user moves their head towards the marker, the radius shortens and the user rotates on a smaller radius. This has the effect that the marker moves towards the center of the FOV, at which point the marker and orbital radius are once again co-located. Figure 4.23 (a) illustrates how the orbital motion resulting from such a process creates a smooth outward spiral trajectory from the current location and orientation towards a circular orbital trajectory with the new marker location at the center. In the reverse case, in which the user rotates their head away from the marker, the radius is extended with the result that, again, their trajectory smoothly transitions, this time via an inward spiral trajectory towards a circular orbit with the marker at the center, as illustrated in 4.23 (b). At every instance, the user's facing direction is exactly perpendicular to their velocity.

Since the marker could be anywhere in the FOV, we would like to independently change both horizontal and vertical orbital curvatures of rotation in order to allow the smooth transition to any new perspective and orbital center. In order to achieve this, then rather than a sphere, the orbital trajectory in general must be on the surface of a torus, whose radii are different in horizontal and vertical directions, and whose axis of symmetry is either horizontal or vertical depending on the ratio of the radii in horizontal and vertical directions.

By calculating the torus surface based on the current relative positions of both the user and the orbital marker, we can thus allow the user to smoothly move from orbit about one center to an orbit about any POI in the user's FOV, as shown in figures 4.21 and 4.22. The user only has to move the marker to the POI, and which ever way they turn will produce the optimal smooth path to any perspective they choose about the new orbit center. In the next section we outline how this process is implemented in an algorithm to calculate the new position of the user in the 3D environment at each frame.

There are several key considerations with regard to creating an algorithm that implements the process described in the previous section. A new position $\mathbf{P}$ in each frame is calculated based on the following initial parameters: $\mathbf{P}_0$ the position of the user in the last frame; the azimuthal, ϕ_0, and zenithal θ_0 angles defining the orientation of the user in the last frame; $\mathbf{M}$ the position of the orbital marker relative to the user and; the current orientation of the user, ϕ_1 and θ_1. In addition, the fixed parameter a controls the pace at which a rotation will cause the orbital distance to reach an equilibrium with the new orbital center (we use $a = 2$).

In the following, x, y and z refer to left-to-right horizontal, down-to-up, and straight-outward directions relative to the user's current orientation (ignoring tilt about the z axis), and X, Y and Z are right-handed world coordinates with Y in the upward direction. In addition r_x and r_y refer to radii of the movement curvature in x and y directions. The algorithm for calculating the user's current position is as follows:

1. Determine whether the user's head movement relative to the last frame is

towards or away from the orbital marker in the x direction.

2. Calculate the radius of curvature in the x direction r_x: In the case that the user is rotating towards the marker,

$$r_x^{\text{towards}} = M_z - a|M_x| \tag{4.1}$$

where M_z and M_x are components of $\mathbf{M}$ in z and x directions. In the case that the user is rotating away from the marker,

$$r_x^{\text{away}} = M_z^2 / r_x^{\text{towards}} \tag{4.2}$$

We also constrain r_x such that $0.2 < r_x/M_x < 5$ to limit the maximum velocity and remove large accelerations.

3. Repeat steps 1 and 2 for the y direction.

4. Find the center of the torus on which we wish to move. In the case that $r_x > r_y$, we can consider a position $\mathbf{T}(\theta, \phi, r, R)$ on the surface of a torus whose symmetry axis is along Y and whose center is at the origin, defined by

$$\begin{aligned} T_x &= -\big(R + r\cos(\theta)\big)\sin\phi \\ T_y &= r\sin\theta \\ T_z &= -\big(R + r\cos(\theta)\big)\cos\phi \end{aligned}$$

where T_x, T_y and T_z are the components of $\mathbf{T}$ in X, Y and Z-axes, r and R are the torus minor and major radii, and θ and ϕ are zenithal and azimuthal angles relative to world coordinates. The center of the torus on which we wish to move is thus given by

$$\mathbf{T}_0 = \mathbf{P}_0 - \mathbf{T}(\theta, \phi, r, R) \tag{4.3}$$

with $\theta = \theta_0$, $\phi = \phi_0$, $r = r_y$ and $R = r_x - r_y$.

5. Finally, calculate the new position, which is given by,

$$\mathbf{P} = \mathbf{T}(\theta, \phi, r, R) + \mathbf{T}_0 \tag{4.4}$$

this time with $\theta = \theta_1$, $\phi = \phi_1$, and again with $r = r_y$ and $R = r_x - r_y$.

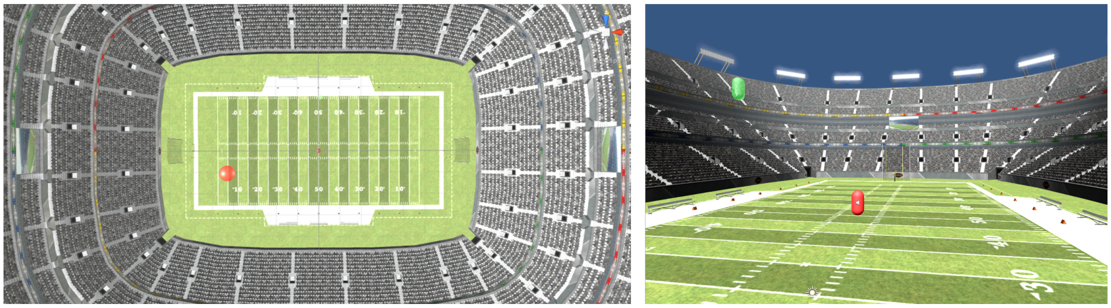

Figure 4.24: The virtual environment used during "HMD Mouse" control scheme user studies. Top: A top down view of the stadium environment. Bottom: A typical view of the user in the VE. The green and red objects are used for the navigation task.

6. In the case that $r_y > r_x$, follow a similar process as in steps 4 and 5, but instead consider a torus whose axis of symmetry is in the horizontal x axis of the previous frame. Since the torus is perpendicular to the forward-facing direction of the last position and orientation defined by ϕ_0 and θ_0, the center can be trivially found by extending this vector direction out from $\mathbf{P}_0$ by a distance of r_y. Then for step 5, define a torus in world coordinates with symmetry axis along X, substitute $\theta = \theta_1$, $\phi = \phi_1 - \phi_0$ and rotate the resultant $\mathbf{T}$ about the origin by ϕ_0.

It is helpful for user control that the marker always be not too distant and in most cases visible [44], and so we recommend limiting its position, depending on the environment context, to for example $M_z < 100$m. If the marker is outside the FOV, we constrain $r_y = r_x = 0$.

If using a head-tracked HMD, the technique can work taking only the 3 rotational DOF as input to AnyOrbit, and so can be used in current 3DOF tracked mobile VR headsets. If available, the 3 translational DOF could be ignored, but we found it more comfortable to allow the user free translational movement relative to the orbital center. To achieve this, record the translational movement of the camera since the last frame and add it to the position in step 5.

There are 4 key control inputs of the AnyOrbit system: the zenithal (pitch) and azimuthal (yaw) angles, the POI (desired orbital center) and the desired orbital radius. In most use-cases explored here, we couple the angles to the corresponding head rotation angles as in previous work [25, 75]. Control of the POI and orbital

Table 4.1: Various control schemes we have tried using AnyOrbit

Configuration Name	*Pitch*	*Yaw*	*Desired orbit radius*	*Desired orbit center*
HMD Mouse	head pitch	head yaw	mouse wheel, left and right click	mouse x, y, wheel
HMD Directed	head pitch	head yaw	Chosen by director	Chosen by director
Desktop Eye-gaze	mouse y	mouse x	Fixed	Eye-gaze x,y
HMD Eye-gaze	head pitch	head yaw	Fixed	Eye-gaze x,y
HMD 360 Video	n/a	head yaw	depends on video	Gaze-selected

radius however can be achieved with a variety of inputs, including by the user with a mouse or eye-gaze, or be controlled by a director who guides the user between different POIs. Table 4.1 summarizes the control configurations we have attempted, and here we will describe user experience and some studies that we have conducted thus far.

HMD Mouse: Head Rotation with Mouse Control

In this case, the orbital marker's position in the FOV is controlled via a mouse input. The marker's distance from the user is controlled using the mouse scroll wheel, and mouse $x - y$ control the azimuthal and zenithal angle to the marker in the user's FOV. The desired radius is set to the current distance to the marker, and so is effectively controlled by the mouse scroll wheel. In addition we provide a function to 'teleport' instantly 50% of the way towards or away from the orbital center using left and right mouse-button clicks respectively. Teleporting instantly in the radial direction avoids a mismatch between vestibular and visual cues known to cause simulator sickness, while a value of 50% was felt allow a satisfactory shift in movement while not being so large as to confuse the user as to the context of their new position relative to the POI.

In user studies, it was demonstrated that within a sports-spectating context,

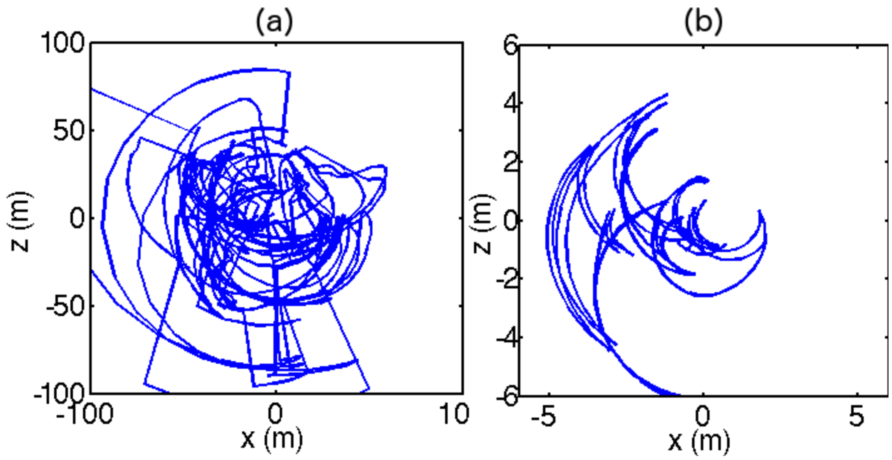

Figure 4.25: Shows position data or the user in the VE while using AnyOrbit, as viewed from above, using (a) HMD with mouse input control, and (b) HMD with directed movement control.

the technique allows smooth shifts in perspective at a rate comparable to broadcast sport, is fast to learn, and is without excessive simulator sickness in most users [98]. A study of 13 inexperienced users (5 female, age $= 25.9 \pm 3.2$) evaluated simulator sickness and performance. Users were tasked with navigating between positions and facing directions in a VE (figure 4.24) representing camera angles typical of broadcast sport, in 3 trials of 5 minutes each. Users averaged 14 ± 8.6 seconds per task. In addition, an expert user completed tasks in 3.8 ± 1.2 seconds. This compares to 5 to 11 seconds for average shot lengths in sport broadcasts. There was a measurable increase in Simulator Sickness Questionnaire (SSQ) scores after the first 5 minute trial, but no significant increase on subsequent VE exposure [98]. Figure 4.25 (a) shows how the user makes good use of the space while moving in circular and spiral arcs.

While we found our initial implementation with mouse input control was effective, we were interested in providing more intuitive and hands free navigation. Therefore we also tried the following control schemes.

HMD Directed: Head Rotation with Directed Position and Radius Control

3D film and storytelling often has the problem that the director cannot control in which direction the user is looking. StyleCam proposed a system in which navigational control is shared between the user and the content producer in order to direct the user experience [16]. With AnyOrbit, the director can control the POI that the user is facing, while simultaneously giving full rotational control to the user.

We created a VE consisting of a sample of 27,000 stars taken from the HYG Database [92]. Figure 4.26 shows the VE, in which the stars' positions, colors, brightnesses and velocities are rendered using aesthetically chosen scaling factors. In this case, we predefined the desired POI and orbital radius. Once a user wants to move on, they can trigger a change to the next predetermined POI and radius by aligning the current orbital center with a designated point. User position data is shown in figure 4.25 (b). The predetermined POIs and radii were selected to give a variety of perspectives, from both within the field of stars, and looking in from outside. Navigation and observation were reported to be instantly intuitive, with one user remarking that it *"feels very dramatic and gives a heightened sense of perspective"*.

We have also attempted to use the same environment and control scheme using the Hololens augmented reality AR headset. The technique seems promising for augmented reality situations, but the Hololens' limited FOV was detrimental to the user experience, since AnyOrbit tends to use the whole FOV.

User Study

When we normally rotate our heads in the real world, the resultant optical flow in the visual field naturally does not cause us motion sickness. We suggest that by moving the rotational center to an exocentric location within the VE, our existing proprioceptive cues can be leveraged to make sense of lateral translational as well as rotational motion, thereby reducing simulator sickness compared to movement that is unrelated to head rotation.

To test this hypothesis, a simple VE was developed in Unity (figure 4.27) and displayed using a 6DOF tracked HMD (HTV Vive). We used a machine running Windows 10, Intel Core i7-3770 and GTX 1060 graphics card. The VE ran at a smooth 120 fps throughout experimentation. We wanted to simplify the movement down to as few degrees of freedom as possible, while not compromising on allowing the user as much freedom to look around the environment as they

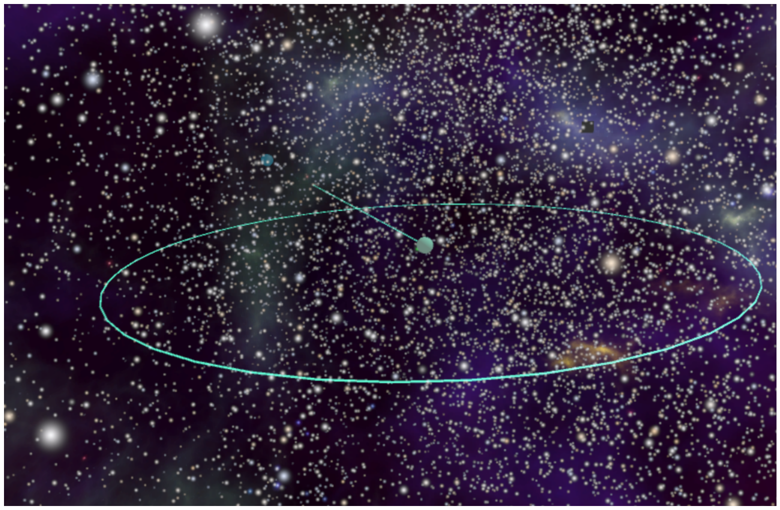

Figure 4.26: User perspective of a VE consisting of stars. The user is guided on an orbital viewing path to different points of interest (POIs) (see accompanying video). The green object in the center identifies the POI and the circle marks the desired radius. Aligning the POI with the distant object advances to the next POI.

wished. We limited user's movement to a straight line of distance $d = 8$ meters. In the control condition, movement was controlled using an XBox 360 controller joystick input, while in the test condition, the position p_x along line controlled by the function $p_x = d\sin^2(\alpha/2)$, which represents a single dimension of movement of a circular orbital path.

16 participants (6 female, 22 to 29 years old, mean 25.3 $\pm$ 1.6 years) took part in the study. Participants were separated into two groups, half receiving the control condition (joystick condition) first and the other half receiving the test condition (head-coupled condition) first, of which they completed one session each. User's were exposed to the conditions on consecutive days or after a minimum of 2 hours rest to minimise any effect of exposure to the previous test condition, and completed the now-standard Simulator Sickness Questionnaire (SSQ) before and directly after each session [70]. Participants scoring a sum of 5 or more on the pre-weighted symptom variable scores (each on a 4-point scale from 0 to 3) did not continue the study and were not included in the analysis, of which there were 5. Participants were asked to remain standing for the 10-minute duration of each session, which allowed a greater freedom of head and body movement. During

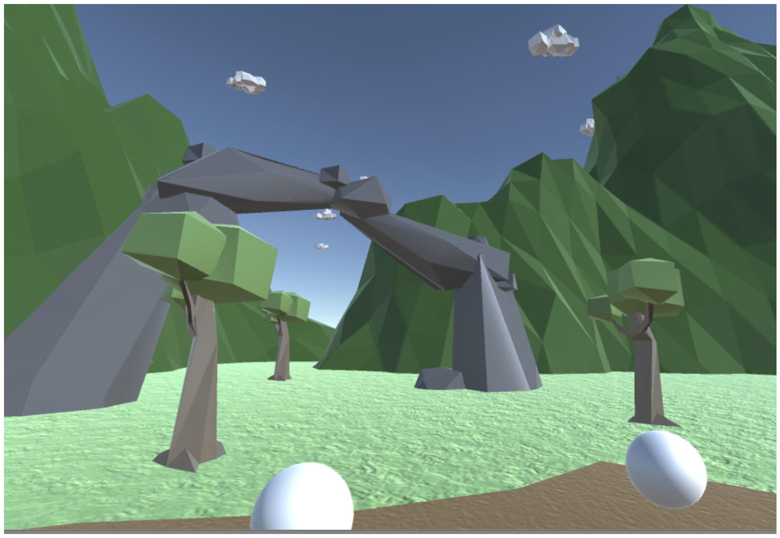

Figure 4.27: User perspective of the virtual environment used in the simulator sickness study.

the session users were asked to move from one end of the range of movement to the other end every 3 seconds, indicated by an audible beep, in an attempt to standardize the average velocity and distance traveled between trials. After the final session, users were asked to comment on the two types of interaction.

Figure 4.28 shows mean simulator sickness scores corresponding to Nausea (N), Oculomotor (O), Disorientation (D) and Total Sickness (TS) scores for the 11 participants that took part in the study. While the means appear to support the hypothesis, scores were highly variable between users and did not adhere to a normal distribution. We therefore used a non-parametric Mann-Whitney U test. A one-tailed paired-samples Mann-Whitney U test showed that the Joystick condition caused a significant increase in SSQ score compared to pre-exposure (U-value = 5.5, p = .00017). The head-coupled condition also significantly increased SSQ score (U-value = 33, p = .038).

We then compared SSQ scores for each condition directly using a two-tailed paired-samples Mann-Whitney U test. The head-coupled condition caused significantly higher SSQ scores than the joystick condition (U-value = 22, p = .012).

Incidentally, while we did not make any particular predictions or design the experiment to test the case in which head-rotation is coupled to forward and backward movement, our simplification of the movement down to a single degree

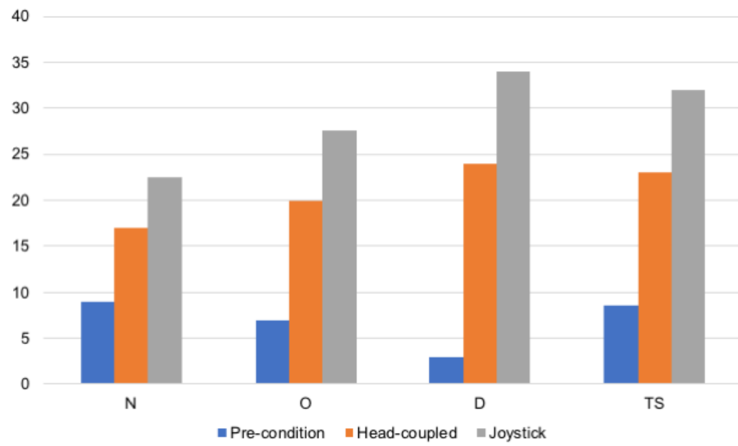

Figure 4.28: Mean scores for each of the three components of the SSQ as well as the total score (TS).

of freedom inadvertently introduced a small amount of such movement into the test condition. Anecdotally, this forward and backward coupling to head rotation was reported to cause a greater sense of simulator sickness compared to lateral movement.

Feedback from participants suggested that people preferred the head-coupled interaction. One participant commented that it felt more natural. Another user commented after doing the control condition first that they developed strategies to combat the nausea induced from the movement using the joystick, *"I felt that moving my head in sync with the controller helped in keeping the balance."* he said. He later told us after doing the second session with the test condition, *"the previous one made my legs feel tired, but this one didn't make me strain my legs"*. Another user reported *"I think I like trial B with head movement better. I felt I have more control."* Finally, a user commented *"Feelings of fatigue less with joystick but dizziness higher."*

This study suggests that coupling head rotation to movement is an effective means to allow movement in VEs while mitigating simulator sickness.

Discussion and Limitation

The AnyOrbit technique allows fluid movement between different orbital centers, and combined with eye-tracking or director control of POIs, allows hands-free interaction that is surprisingly intuitive.

A problem with the current implementation is that it is easy for user's to travel into and through virtual objects, which is known to be disorientating. There are a variety of mitigation strategies for this problem in the literature [44,83,115], some of which could be employed in combination with our system. In particular, the orbital center could be shortened as a user approaches an object interface, which would have the automatic dual effect of reducing their velocity and steering them away from virtual objects.

Recently FOV restrictors were found to reduce simulator sickness and may be recommended for large accelerations or velocities when experiencing VEs using HMDs [42]. Such FOV restrictors may be particularly suited to our system, especially when using our eye-tracking schemes, because large accelerations naturally occur when the user's gaze is in the center of the FOV and the orbital radius is maximised.

4.4.2 GazeSphere

This subsection is adapted from a poster paper presented at Siggraph 2017 [103]. The paper was co-authored by myself, Benjamin Outram, Benjamin Tag, Megumi Isogai, Daisuke Ochi and Kai Kunze. This work was also a collaborative effort and funded by NTT Media Intelligence Laboratories. We next look at the multimodal input of combining head movement with eye gaze for a convexed interaction in the intimate space. Head movement consumes extremely limited proxemic space, whereas eye gaze does not use any additional proxemic space at all. We developed GazeSphere, a system that utilizes both of these inputs to be used for navigating 360-degree video environments.

Current 360-degree image and video use-cases do not allow for a smooth transition between several environments, often resulting in a jarring experience. For example, programs like Google Street View, which employs 360-degree images of streets and buildings, simply teleports users to the next desired point of interest. Teleportation, though effective, has two fundamental problems; it is unrealistic and breaks the sense of immersion, and it requires a physical input such as a mouse click to move to the next point. Most other applications for navigating 360-degree environments do not solve the first issue, however, The second issue is often solved using 'dwell time', where the user is required to face the desired direction for a period of time to activate a transition. This method which works well for allowing the viewing of the environment in VR while also providing a hands-free solution that is unobtrusive and subtle [99]. However, herein also lies a new challenge; given a smooth transition between two points is possible, then the ability to stop at any point to view the environment during the transition is also favorable. Ideally, a hands-free input method is needed as a substitute for physical buttons for this function. One of the more popular solutions right now is eye tracking. Despite not being readily available for the average consumer at this point in time, eye tracking remains of interest to researchers and developers of VR technology. For example, Transparent Reality [101] and GazeSim [104] utilized eye tracking in VR as an input modality by computing the focus depth of the eye gaze to transition between layers of information and for foveal rendering respectively. Layered Telepresence [131] also shows the promise of eye tracking for use in simultaneous multi-presence using multiple telepresence robots. In terms of transitions in virtual environments, there have been several solutions that have been developed for locomotion, but most of them are obtrusive, requiring physical buttons or wide gestures. an interesting method was proposed in AnyOrbit [98] that uses head rotation and exocentric rotation for locomotion that minimizes motion sickness. However, this method required a separate mouse input to allow user control of the center of rotation.

Implementation

The GazeSphere system was developed with several goals in mind, one of which was to limit the overall input devices and create a hands-free solution to navigation between stationary positions represented to the user through 360-degree video. To achieve this, we used the Pupil Labs' eye-tracking infrared camera system that has a high sampling rate of 120Hz. The user first must undergo a short calibration phase by looking at virtual markers placed at the edges of the display, close to the periphery of the user's vision. The 2-dimensional gaze position can then be computed, and a ray is calculated from the user's position towards the gaze point. For capturing 360-degree video, we use both the GoPro Omni for 4K video recording and Ricoh Theta S for prototyping.

Our implementation utilizes head rotation for transitioning from one point to another [98]. The system uses 360-degree videos taken at two stationary locations, and a 360-degree video shot continuously between these two locations, whose position moves along a single direction between the two points at a constant speed. The video's key frames are maximized to allow fast seeking both forward and backward in the video. We then seek this transitioning video using head rotation, where rotation of the head left and right is coupled to seeking forward or backward in the video. Since the video is shot at a constant velocity, time in the video is proportional to distance, and so rotation of the head gives a sense of linear locomotion in physical space. This allows a more orbit-like sensation and intuitive understanding of the beginning and end points of the transition. To maintain hands-free interaction and allow the user to select from among multiple possible transition directions, we employ real-time eye tracking as input. Virtual cue objects are placed relative to the user in the direction of possible transitions. For example, a virtual cue is placed at the end of a street where the user wishes to go. If the user looks at the virtual cue, the transition is initiated and the user can navigate toward that direction using their head rotation. During this phase, we employ AnyOrbit's torus-based orbit algorithm, which produces a self-correcting orbital path of the user around the cue object, and enhances the orbital motion metaphor that is very intuitive to users. While transitioning, the user can, at anytime, deactivate orbital motion by looking away from the cue object, thus returning to a stationary egocentric rotation view. This system provides the user with a great level of freedom when exploring 360-video environments, providing the ability to navigate to multiple points of interest, as well as any point during transitions between points of interest, while at all times remaining hands-free. Figure 4.29 illustrates the arc of the orbital motion.

User Study

Initial studies and demonstrations have shown that users were able to quickly adapt to this motion without any noticeable motion sickness. Mapping head rotation to linear movement is novel but intuitive to most users. For example, most users were able to master a related navigation technique in 5 minutes [98]. GazeSphere was demonstrated at the Dagstuhl Seminar 2017 entitled "Beyond VR

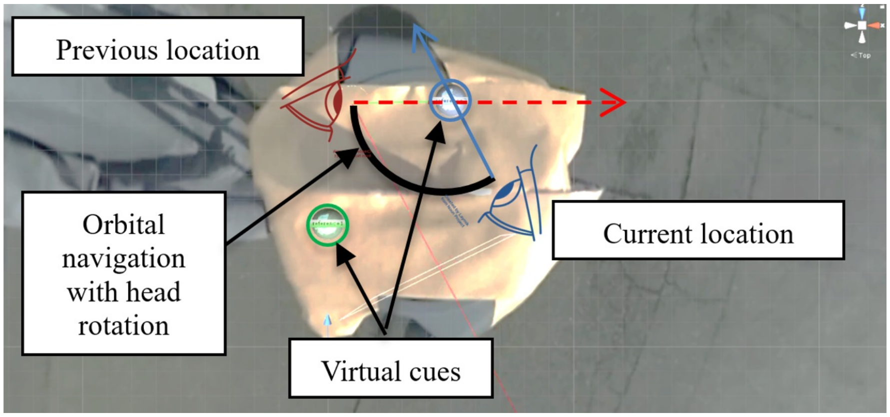

Figure 4.29: System overview during navigation

and AR: Reimagining Experience Sharing and Skill Transfer Towards an Internet of Abilities." Most of the participants consisted of senior researchers from various Computer Science and Engineering Research Laboratories. Feedback gathered was used to improve the system, such as making the environment slightly transparent during transitions to minimize motion sickness.

The proposed system can redefine navigation mechanics in HMDs for 3D content as well. For example, Figure 4.30 shows the navigation mechanic being used for CAD modelling, sports viewing, data visualization, and gaming. Orbital navigation allows intuitive viewpoint selection to allow, for example, freedom to observe particular sports plays in sport, or particular data points in 3D data visualizations. We have demonstrated a game utilizing this mechanic that requires players to position and align virtual objects through orbital motion and exocentric rotation. Finally, the most obvious application would be for virtual touring and street view-like applications that allows the user to move to points of interest intuitively.

Discussion and Limitation

AnyOrbit, and its variation GazeSphere, both show that the human's cognition is so flexible to the point that, even under two extreme circumstances that differs from the human norms, users are still able to adapt to it in a short period of time. The first extremity is to limit the movement to only within the user's peripersonal

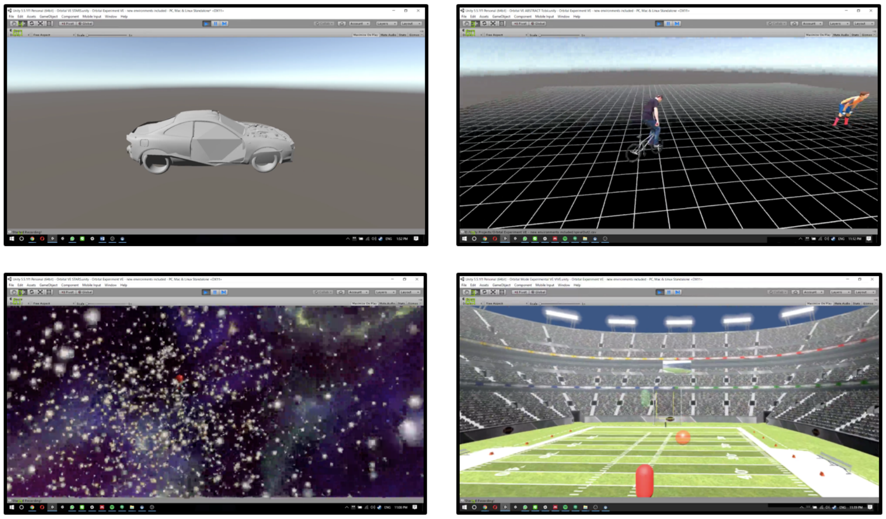

Figure 4.30: GazeSphere used in 3D content like (a) CAD modeling, (b) sports viewing, (c) data visualization, and (d) gaming

space, whereas the second is the alteration of the output of the interaction that differs greatly from the norms. This also brought with it an added benefit of reduced motion sickness.

4.5. Convexed Selection and Activation: Eye Gaze

For the next two studies, I investigate the selection and activation interaction type, which are serial-based interactions. Eye-EMG looks into a multi-modal approach, whereas Transparent Reality is a single input system based on eye-gaze detection only.

4.5.1 Eye-EMG

For exploring selection and activation, I applied a mutilmodal physiological sensing method by combining eye gaze with muscle contraction. Since very little motion is noticible, this also falls under peripersonal space. The mapping type is shortened to that of only looking at an object, and flexing the forearm muscle, that covers both arm movement and finger grasping of an actual pick-and-place. For the latter, the motion type is different since the user only needs to flex the forearm muscle, though the expected output is linear.

VR as a platform is increasingly immersive as they began with tracking the user's head direction, and moved towards tracking their hands' position, to allow for various forms of gesture and button-based inputs. This has become the conventional input system, yet also the current limitation for VR; the use of positionally tracked physical controllers that are often seen with a directional pad for analog input and various buttons for each finger for digital input. Though reliable, we wish to delve deeper into the next step of input for VR. We propose the use of physiological signals as an additional input method and to enhance the overall experience. We chose to leverage the use of eye gaze tracking and muscle sensing as a form of selection and activation based on several reasons. Firstly, we wish to investigate hands-free methods of input to complement conventional hand controllers that currently exist today. Secondly, both of these signals allow reliable explicit input compared to other forms of signals like electroencephalogram (EEG) and electrocardiogram (ECG) that relies on brain activity and heart rate respectively.

Our approach is based on related research in VR interaction techniques, physiological sensing, and select-and-point task assessments. For VR interaction mechanics, we look into non-conventional input and sensing mechanics for VR like gesture-based detection and peripheral-based input devices. Physiological signals though, have been gaining popularity with several related work in VR. We then look into select-and-point tasks in VR and how they are evaluated in terms of accuracy and performance.

Several researches in VR introduced novel methods for manipulation. For example, a spherical manipulation device was made using a tilt sensor, electric compass, and flex sensors that allows gesture recognitions. This implentation combines a peripheral with gesture recognition, though the study was conducted only for spherical interfaces [80]. This work is similar to the AcceleGlove, which presents a whole hand input device using accelerometers [53]. The device was developed to recognize sign languages, and can also be used as a substitute for a computer mouse for selection and activation. Another work combined gaze with gesture using a Microsoft Kinect with Mirametrix tracker for desktop use [138]. Though no user study was conducted, it served as a proof of concept for a multimodal-based input that may increase effectiveness of interactions in virtual space. Other peripherals that were developed catered more towards a specific use of input, such as locomotion. For example, the omni-directional threadmill was developed for training the army in virtual environments [32].In this regard, it was extremely effective, however the cost and overall size of the device prevented it from entering the mainstream market. Another work used a Wii balance board as a low cost input device [164]. Besides locomotion, it could also be used for 3D object manipulation and application specific task due to its discrete and continuous signals.

The usage of EMG may not be mainstream yet, but its feasibility as a daily interaction device is still highly promising. EMG has been used in various medical applications such as muscular rehabilitation, muscular disease, and prosthetics control [9, 52, 62, 72, 90]. During a feasibility analysis, it was found that an accuracy of up to 95% was achievable, implying that EMG interaction can be highly promising, especially given its discrete nature [90]. EMG has the potential for gesture-based recognition such as fingers, hand, and arm motions. However, it was found that in terms of accuracy, EMG performs best simply with direct muscle contraction. Activities recognition like carrying a heavy bag or mug has a lower error percentage compared to recognition of pinching gestures [128]. Furthermore, precise calibration is required for accurate gesture recognition due to the data being extremely user dependent. A previous work used gesture-less EMG for various inputs by determining the length of muscle contraction time, however, relying solely on EMG greatly limits any form of selection [30]. Using length of contraction also suffers from the same issue with dwell time. One of the solutions for this is to pair EMG with other forms of input, thus creating a multimodal system. Multimodal inputs are no stranger to human-computer interface [171], and can certainly improve an interaction mechanic if done correctly. For example, coupling EMG with a touch screen allows the system to recognize which finger is touching the surface, as well as the amount of pressure exerted through muscle sensing without the use of a pressure sensor [11]. Besides EMG, other forms of multi modal input like combining eye gaze with hand gestures have also been explored [20].Regarding eye tracking, it is a sensing mechanic that is very likely to be adopted into future VR headsets. As of this moment, companies like Tobii have been active in the eye tracking scene, though in the case of VR, most solutions are still extremely expensive or not made available for consumers yet. In research, eye tracking has been used for foveated rendering [104], multi-user scenarios [131], or other forms of input modality. After a quick calibration, it is an effective tool that adds an additional layer of interaction, making it a perfect addition for VR. Coupling eye gaze tracking with EMG is a novel proposal and useful input, seeing as how it has previously been used for motor disabilities [23]. This proves that such a multimodal input can be beneficial given the right context.

Evaluating the performance of a new input method, whether it be for AR, VR or any platform, requires the correct assessment method. A simpler method would to simply measure the time required to complete a certain task, such as walking from point A to point B using the assigned input method [17]. However, a more through evaluation can be seen in text entry studies such as work related to keyboard input [82]. In a HoVR-type where the user developed a system for smartphone keyboard input in VR, they measure the time and accuracy to complete a given phrase or sentence, coupled with a general usability questionnaire [73]. One of the more common evaluation method is the Fitts' Law method for relative measure of performance, also used in a study that combined gaze detection with gesture [20, 82]. Essentially, it is a selection and activation task for the user that includes a proposed solution being compared with a baseline. This was also utilized in a study to compare between several gamepads [119].

Based on the related work, it can be seen that there exists many forms of input and interaction techniques utilizing sensors, gestures, and currently available devices. Among these input methods, use of physiological signals carry the benefit of being intuitive and more accurate depending on context. To evaluate this, utilizing Fitts' Law is appropriate as it is a well recognized evaluation method, though an additional user experience study would benefit for an additional qualitative feedback.

Implementation

The core interaction method proposed in this multimodal solution is to have the eye gaze used as a selection mechanic while the muscle contraction used as an activation. This can be similarly dubbed as the eye gaze being part of the target acquisition phase and the muscle contraction as target action phase [20]. One of the core benefits as opposed to dwelling techniques is that the user is not required to focus on an object for an extended period of time which is fatiguing [60]. Therefore, only muscle contraction is used for the target action phase. We will perform a comparison by using eye gaze for the action phase as well in our user study.

Special considerations are required when designing an interface that relies on eye tracking. For other forms of applications that are not VR-based, determining the accuracy threshold of the eye tracker is important [20]. However, UI design should be scaled accordingly, which is also parallel with the design consideration for VR UI [154]. Generally, if a UI is too small and requires really accurate eye tracking for precise solution, then it is too small for proper viewing in a HMD. Most UI are directly attached to the player camera, however, this does not apply for spatial UI or 3D objects that are placed in the virtual environment. This is because the user can physically move closer to the object for a better view for VR solutions that use spatial tracking. For the target action phase, two kinds of interaction can be performed: a discrete action in form of a single activation method like a lamp switch, whereas continuous action provides a stream of data for as long as a condition is true, such as holding down a keyboard button.

In this context, discrete action means a single muscle contraction, while continuous action means continuous muscle contraction. This allows for several interaction mechanics for the user in a VR environment. For instance, discrete action is great for locomotion mechanic, and activation of a menu. In a shooting game, this is akin to firing a semi-automatic pistol. Continuous action on the other hand is suitable for dragging objects around like a paint brush, manipulating a graph bar, or firing a fully automatic rifle in VR. These two distinct methods of interaction create a taxonomy similarly proposed by Chatterjee et, al. [20]. More generic actions like pick-and-place or drag-and-drop may freely utilize either interaction mechanic.

All of the peripherals used in this study can be obtained off the shelf. Granted, the accuracy may not be as high as industrial or medical grade tool, but consumer electronics are the more accessible option for regular users. For the VR HMD, we use the HTC Vive VR headset. For the eye tracker, we used the IR trackers by

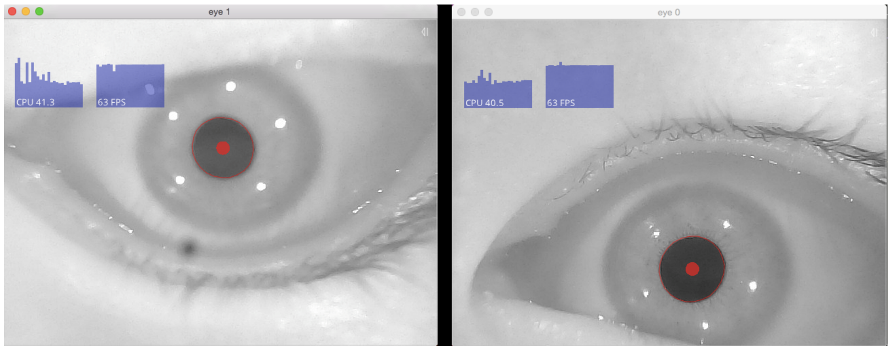

Figure 4.31: **Eye tracking from the Pupil software.**

Pupil Labs. These are a pair of 120Hz cameras that can be bought with direct integration with the lenses of the Vive. Fig 4.31 shows the user's eyes being tracked. We also chose the Vive because it can easily be made into an AR HMD simply by attaching stereoscopic cameras.

There are several eye-tracking solutions for VR, however, they are either extremely expensive like SMI's solution [140], or not available for consumers at this point of writing. Eye tracking solutions are relatively less for AR, though they can be combined between a third party tracker and Hololens. Finally, for EMG sensing, we opt to use the Myo Armband, which contains eight medical-grade EMG sensors placed on the user's forearm [91]. Myo also comes equipped with an accelerometer and gyroscope for gesture input, but for the purpose of this study, we will be excluding these.

In terms of software, we will be interfacing with the Unity Engine due to its seamless integration with VR. C# is the primary development language. To obtain the eye tracking data from the Pupil trackers, we use open sound control (OSC) and ZeroMQ to pipe the data over to Unity, allowing us to both obtain the raw data and toggle Pupil's calibration directly from Unity [130]. For the Myo armband, Myo provided a Unity plugin that directly pipes raw EMG data for each of the sensors. The machine used for this study is equipped with a Core i7-6700 processor, an Nvidia Geforce GTX980 graphic card, and 8GB of memory. It is important the machine used is at least above or equal to the minimum requirement for VR suggested by Oculus [96] to avoid any additional delays that may cause motion sickness.

User Study

The user study is divided into two parts; one for evaluating the performance and throughput, as well as the overall qualitative feedback in a use case scenario. Both parts of the study will be explained thoroughly in their respective subtopics. They first need to perform a brief calibration to map the tracking data into Unity's 3D space, though the gaze data is only in 2D. The calibration is performed by looking at 9 points that appear on the FOV sequentially. No calibration for the EMG sensor is necessary.

A total of 16 participants were recruited for this user study, comprising of 7 females and 9 males aged between 22 to 38 (mean = 26.19, SD = 4.55), where their feedback and opinions were also collected at the end. All the participants had no prior knowledge of the association of EMG and eye tracking with VR, though some of them have experience with interaction in VR environments. For participants who wear glassed, it needs to be taken off in order for the eye trackers to detect the eyes accurately for calibration. Each participant also provided their informed consent, and no identifying information is provided in this study.

We used a Fitts' Law study shown in Figure 4.32 to evaluate the throughput of our proposed interaction based on varying index of difficulties (ID) and input methods. This particular form of study has been used to evaluate multimodal input method before and has been shown to be a viable method for the evaluation of pointing techniques [20]. It is worth noting that Fitts' Law has been deemed suitable for gaze input as well [88,161,165,173], though none has evaluated the use of EMG in a VR environment. We directly compare our method with gamepad input, motion controllers, dwell time, and dwell time by gaze-only. Even though mouse input has been considered for comparison due to most people's comfort in using it, we deemed it unnecessary for VR as conventional VR input leans more towards a gamepad or motion controller. Therefore, the independent variables are the 5 different input types listed below, with 6 levels of IDs (2.81, 2.94, 3.07, 3.2, 3.33, and 3.46 bits). The dependant variables are the movement time (MT) as well as the effective index of difficulty (IDe) to calculate the final throughput (TP). The explanation for each of these variables will be further elaborated in the following subsection.

- Gamepad: The gamepad was the first consumer VR input method before the introduction of motion controllers as most users are accustomed to its layout. It can also be combined with AR systems, primarily for gaming purposes. Therefore, the gamepad's analogue stick is used for selection and a button is used for activation.

- Dwell time: Since one of the primary benefits of our proposed method is that it saves time, it is only fitting to compare it with a dwell time method that is currently being used for most hands-free interaction mechanics. Dwell time places a reticule in the middle of the FOV and requires the user to use head rotation to place the reticule over interactive objects for a period of time, in this case, 750 milliseconds. Dwell time by eye gaze have shown that this value provides the highest throughput for dwell-based interactions [173]. Though this input method does not rely on using eye gaze, a similar dwell time value is used.

- Dwell time with eye gaze: Similar to conventional dwell time implementation mentioned above, this method instead uses eye gaze for selection. As of this point of writing, this method is not popular with consumers simply because eye tracking is not yet made mainstream to the average users. After a brief calibration, a similar dwell time of 750 milliseconds is also assigned.

- Eye gaze with EMG: Our proposed method replaces dwell time with muscle contraction activation. The user simply needs to contract the forearm muscle to enable activation at the point of gaze, akin to clicking a mouse button for where the cursor is placed.

- Motion controllers: Motion controllers are essentially gamepads that are tracked in the virtual space. Since most current VR interfaces rely on them now, we used the conventional "laser pointer" interaction where a raycast is produced from a motion controller to point at selected targets. Activation is achieved by pushing the trigger button on the controller. We deem this input method as the gold standard baseline for VR input and interaction. Such an input is not yet readily available for commercial AR at this point of time.

Among these 5 input methods, only the gamepad requires a preset cursor velocity as other methods depend on the efficiency of the user themselves. The default speed of a virtual object according to the maximum displacement of the analogue stick is 120 units/second. We ran an informal pilot study with 8 participants who are unrelated to the main study, with varying amount of experience using a gamepad, to determine the most suitable speed of the cursor. Each participant is simply required to enter the VR environment and move a cursor between 2 points placed left and right at the opposite end of each other using the gamepad. 6 levels of distances, which were equivalent to the amplitude (3, 3.2, 3.4, 3.6, 3.8 and 4 units), were set, and each participant is given the freedom to adjust the speed of the cursor until they find a comfortable speed balance between the distances. We found that users with little to no experience prefer the cursor to be slower around 96 units/second. because they have trouble positioning the cursor accurately, whereas participants with moderate experience in using a gamepad found that the lower speed is too slow, yet higher speed was very easy to overshoot. However, all participants agreed that cursor speed above 144 units/second was quite easy to overshoot, even at a maximum distance of 4 units. Therefore, we decided to use the default speed of 120 units/second without any speed multiplier.

For the main study, each participant is required to use each input method according to each preset ID, three times per ID. Therefore, each participant will run this study 90 times. To eliminate any ordering effect, the sequence of input method was counterbalanced according to the Latin Square. The sequence of ID was randomized for a more realistic range of difficulty [119].

The ISO 9241-9 Fitts' study method was used for this experiment [1, 20]. According to Fitts' Law, the index of difficulty, ID is influenced by the distance from the center point to any of the targets and the width of the said targets.

$$ID = log_2(\frac{A}{W} + 1) \tag{4.5}$$

Where A is the amplitude, or distance and W is the width of target, in unity units. The targets in this experiment are modeled as green spheres in a 3D environment. When the experiment begins, a green target will be rendered red and the participant is required to activate the sphere (depending on the input method), followed by the next target which is opposite the previously selected target until each of the targets are activated. The trial ends when the last sphere is the same as the first. We also record the movement time (MT) of the whole trial for each input method. Since it refers to the time the user spends moving a pointing device, we exclude the dwell time for dwell-based interactions as it is assumed that the pointer is static. At the beginning of each trial round, the position of the cursor is reset to the middle of scene. At the point of selection, we also measure the standard deviation (SD) of over-shoots and under-shoots from the center of the target spheres. We use the SD values to calculate the effective width, We, shown below:

$$We = 4.133 * SD \tag{4.6}$$

This allows us to compute the effective index of difficulty, IDe, and the final throughput, TP.

$$IDe = log_2(\frac{A}{We} + 1) \tag{4.7}$$

$$TP = \frac{IDe}{MT} \tag{4.8}$$

During the entirety of the study, we employed a think aloud protocol where at anytime, the participants are free to express their opinions and provide constructive feedback, which is recorded. Finally, after each participant experiences all the IDs of an input method, we ask them to answer the NASA Task Load questionnaire for each input to understand their perceived load.

The second study focuses on qualitative feedback from participants where we obtain informal feedback regarding the proposed input method in a simple game [65]. The game places the participant in a room, equipped with a pistol that aims at the direction the participant looks, as shown in Fig 4.33. The participant is simply required to look at targets that spawn in front of them and fire a bullet based on muscle contraction.

To complement the main study, we also calculated the accuracy of each bullet shot, by finding the distance between where the bullet lands and the center point of the target and evaluate the error rate [51]. However, we did not compare this with other input method and simply treat this as a minor performance evaluation.

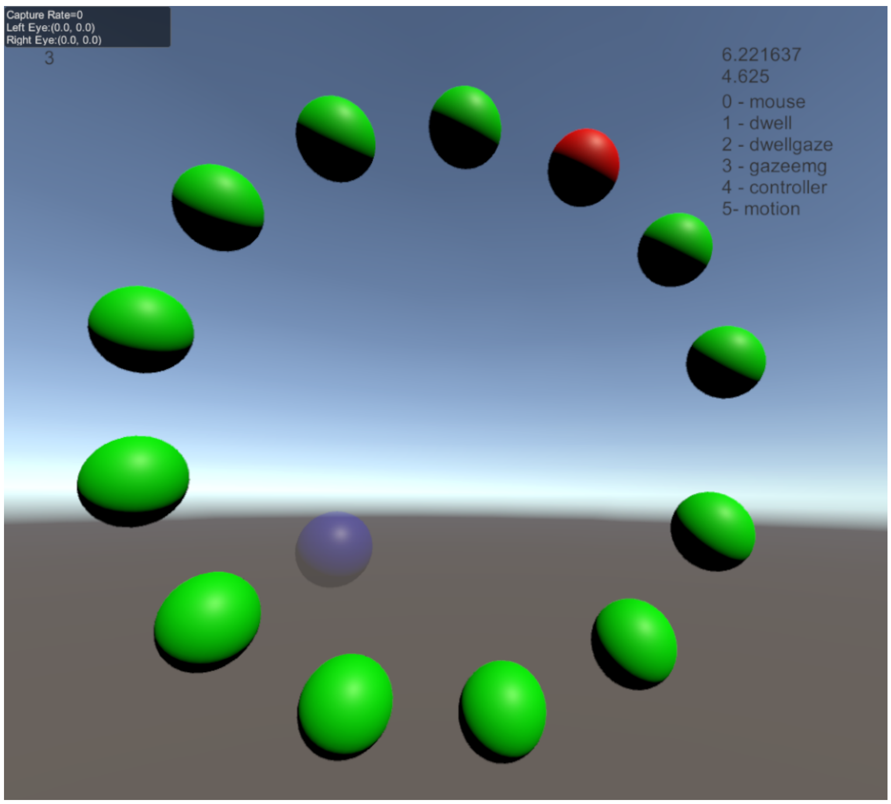

Figure 4.32: **The screen of the user study, with the red point denoting the target**

Figure 4.33: **Shooting game to obtain qualitative feedback, score, and accuracy**

Results

We assessed the Fitts' Law index of performance (IP), recorded subjective work load via a NASA TLX questionnaire, and collected qualitative feedback from a simple shooting game.

We conducted a two-way repeated measure ANOVA with the dependent variable throughput and the two fixed factors target selection method and level of difficulty. An interaction between condition and level of difficulty could not be demonstrated, $F(20, 450) = .241, p = 1.0$. There was a statistically significant effect of the target selection method on throughput, $F(4, 450) = 89.211, p < .0001$. With $\eta_p^2 = .442$ this is a large effect accounting for 41.5% (adjusted $\mathcal{R}^2$) of the variance in throughput. Level of difficulty did not have a statistically significant effect on throughput, $F(5, 450) = 1.433, p = .211$. Figure 4.34 visualize the estimated marginal means of throughput and shows the dependency on the target selection method. Post hoc analysis conducted with a Tukey's range test showed that target selection by motion yielded significantly higher throughput than all other methods ($p < .0001$). Dwell lead to significantly higher throughput than gamepad ($p = .011$) and gaze ($p < .0001$), whereas gaze and EMG resulted in significantly higher throughput than gaze alone ($p = .001$).

To assess the mental work load perceived by participants we calculated the Raw TLX score after each condition and applied a Friedman test on the five control conditions. Here, we found a statistically significant difference between conditions, $\chi^2(2) = 25.509, p < .0001$). We conducted the post hoc analysis with Wilcoxon signed-rank tests applying a Bonferroni correction, which resulted in a significance level of $p < .005$. Median perceived work load scores for Dwell were 40 ($SD = 17.6$), for Dwell and Gaze 51.7 ($SD = 12.3$), for Gaze and EMG 29.2 ($SD = 12.7$), for gamepad 25.8 ($SD = 15.2$), and for Motion 29.6 ($SD = 10.3$) (see Figure 4.35). There was a statistically significant reduction in perceived work load when using Gaze and EMG ($Z = -3.124, p = .002$), the Gamepad ($Z = -3.362, p = .001$), or Motion ($Z = -3.465, p = .001$) compared to Dwell and Gaze. There was no statistically significant difference between the remaining conditions.

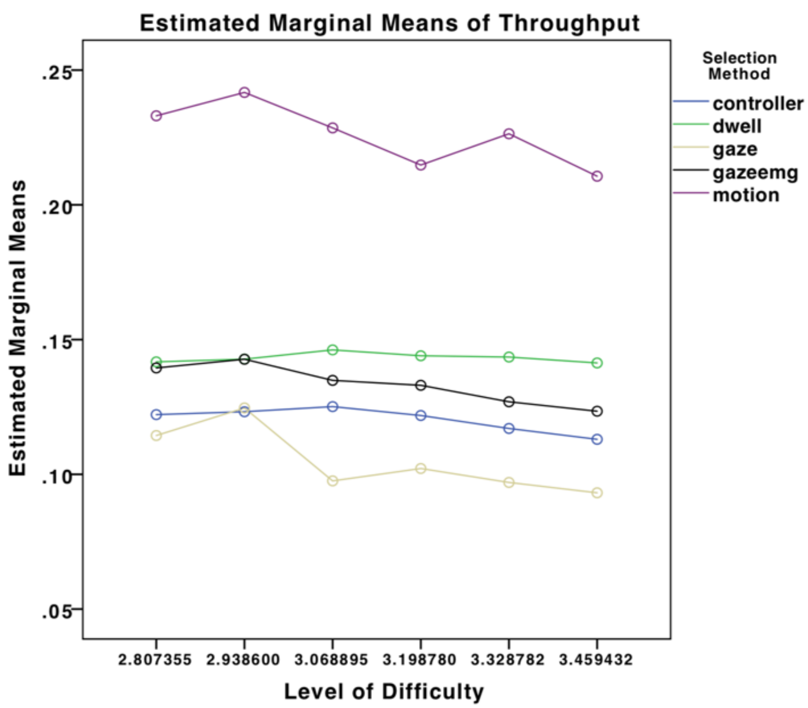

Figure 4.34: Throughput is highly dependent on the target selection method (lines), whereas level of difficulty does not yield a statistically significant effect (slope). There is no interaction effect between the two fixed factors selection method and level of difficulty.

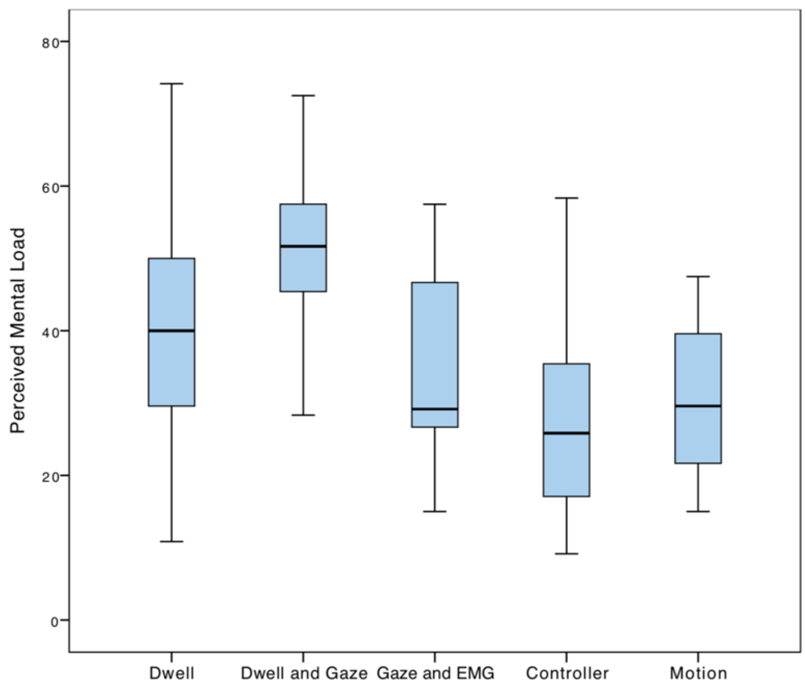

Figure 4.35: Perceived work load as a result of the NASA RAW TLX scores: target selection with Dwell and Gaze was significantly more strenuous than using gaze and EMG, the controller, or motion.

Discussion and Limitation

Overall, we demonstrated the feasibility of combining eye gaze with EMG for VR interactions. Based on the results, we found that dwell gaze performed the worst, which is to be expected because the participants were required to fixate on a point for a period of time, which can be uncomfortable or straining, especially given the length of time and number of trials they were required to do so. However, by removing fixation and replacing it with instant activation using EMG, the performance significantly improves. Using the motion controller still outperforms other input methods which can be due to several factors, namely due to how it feels like a direct pointing gesture, though it is not a hands-free solution and therefore could be more limiting in terms of introducing additional input or for AR and mobile VR solutions. Finally the gamepad was the slowest since unlike eye, head, or arm-based selection, using the gamepad requires a preset maximum speed and thus cannot be as fast as the participant wishes it to be.

When asked to provide qualitative feedback during the Fitts' study, a total of 7 participants preferred to use the combination of eye gaze with EMG, 8 participants preferred using the motion controller, and 1 participant preferred using the conventional gamepad the most. When we interviewed the one participant that chose gamepad input, he or she mentioned that it was the most relaxing and least taxing, requiring near to zero physical movement, thus reducing motion sickness because he or she did not have enough rest the previous night and was feeling slight nausea and disorientation prior to participating our study. For the 8 participants who preferred the motion controller, 3 of them mentioned that it was simply more fun since the virtual laser pointer from the tip of the motion controller looked visually similar to a fictional sword in a popular science fiction movie, thus increasing the enjoyment. It was also overall fun for these 3 participants who had never experienced a motion-based controller before. The remaining 5 participants who chose motion controller was simply due to its balance between speed and stability. For lower ID, a wrist movement was enough to perform selection, though at higher ID, arm movement was necessary thus increasing fatigue over time. Furthermore, all 8 of these participants experienced some difficulty in calibration for the eye tracking, thus reducing the overall score of both the eye gaze dwell, and eye gaze with EMG. For the participants who preferred eye gaze with EMG, 5 of them stated that it was extremely fast, since selection was achieved using eye gaze which feels natural, and activation was using arm muscle without much force, thus negating any fatigue. The other 3 participants enjoyed the novelty of the interaction over other more conventional methods, since these participants also have moderate to expert experience in VR interactions.

Among other comments, one of the participant mentioned that the loading time used for dwell and dwell gaze is frustrating. Another participant had no issue with the loading time, but dwell solution was tiring for the neck, especially for higher ID. A participant who tried the gaze solutions first followed by dwell, found that he or she felt strange that activation did not start after looking at the points with the eye, possibly due to ordering effect.

One of the main limitations at this point of time is in fact the accuracy of eye gaze tracking. With precise calibration, it is possible to obtain near perfect tracking. However, in most cases, participants need to calibrate several times to achieve the desired level of accuracy. Furthermore, eye tracking is a very delicate method that uses IR cameras. Occurrence of slippage in the HMD after calibration will completely render the tracking unusable and further calibration is required. This means that the user cannot make any sudden or extreme head movement to maintain eye tracking precision. However, it is worth noting that as this technology becomes more mainstream, the tracking will certainly improve. Another common issue was that since the position of the trackers are fixed in the HMD, it largely depends on the participant's compatibility with the trackers. From our study, we found that the eye trackers were difficult to detect the eyes of Asian participants, whereas seems to work relatively well for European participants. This can be due to several factors such as eye color, size, and relative position of the eyes. As of this moment, the currently used setup is suitable for our user studies, however, a separate MacBook was required to operate the eye trackers. Nevertheless, we believe eye tracking will be the next evolution for VR and AR technology once it matures. Regarding the utilization of EMG, since minor muscle contraction can be easily detected by the sensor, it is possible for accidental activation to still occur. This is a common problem for input methods that do not involve buttons. Nevertheless, software tweaks on the UI behavior or increasing the activation threshold can easily circumvent this issue. High muscle contraction only even happens if the user is lifting something heavy or performing an extreme motion like punching, unless done on purpose. Comparing to interactions like arm or finger gesture, or pressure sensors on certain body parts, EMG activation is less likely to accidentally occur.

Compared to the previously studied parallel-type interaction (locomotion), serial-type interaction (selection and activation) has several motions being done procedurally, and this study is meant to understand the effects of convexing this motion. We can now see that it is entirely possible to still create a more efficient form of interaction for this, since the proposed EyeEMG method is able to perform nearly as well while taking much less physical space.

4.5.2 Transparent Reality

This subsection is adapted from a poster paper presented at the ACM User Interface Software and Technology Symposium (UIST 2016) [101]. The paper was co-authored by myself, Benjamin Outram, Noriyasu Vontin and Kai Kunze. This work was also a collaborative effort and funded by Fujitsu Design. In this next approach, the procedure of interaction is rearranged so that the final desired output (object picking or activation) is linked to the very first input (looking at the object). Unlike finger grasping though, the activation mechanic here uses the change in focus depth, making it a different motion type that creates a non-linear output that is more suitable for toggling interfaces in different depths.

Due to the versatility and usefulness of eye tracking, it is without a doubt an extremely plausible evolution of the next iterations HMD, judging by the already available Fove headset [45], SMI [140], and Pupil eye trackers [130]. However as of this moment, the instability in calibration and overall cost for the trackers still prevent them from mass adoption, though this is simply a matter of time.

Implementation

We have also developed an early prototype for eye tracking last year in a collaborative project with Fujitsu design. The first eye tracking-equipped DK2 was made through a slight modification of the lenses shown in figure 4.36. Its main benefit over Fove is that 1) they are still not available at the point of time where

Figure 4.36: Placement of the eye trackers in DK2

this prototype was being developed and 2) they only work on the Fove headset and are not compatible with other readily available HMDs. Our prototype carries three main advantages; 1) It was compatible with the DK2, which was the most popular HMD at that point of time, 2) it provided high quality eye tracking with a frequency of 120Hz at the fraction of a cost (roughly $2,000) of SMIs solution, and 3) it provides the user with all the necessary tracking data, including pupil diameter, confidence, and the directional vectors per eye. The eye trackers that were used were the Pupil Labs tracker, which at that time, was sold as a separate wearable and was not VR compatible. To make them compatible with the DK2, about 1/3rd of both lenses were modified and cut to create enough space for mounting the IR cameras. It was found that despite the modifications, it did not obstruct the users experience in consuming the VR content.

We look at 2 separate implementations and use case for eye gaze tracking for virtual environments; one as a selection modality, and the other as a solution for foveal rendering. Eye tracking and eye movement analysis is often used in psychology experiments, marketing etc. to better understand users intentions [3], as implicit input in gaming or as automatic tagging and context recognition tool during everyday life [58]. So far, there are only few researchers exploring explicit eye gaze based interactions , as users often feel eye fatigue [36]. The best approaches seem to use some stimulus (e.g. smooth pursuit) for less stressful gaze interactions [40]. This paper presents an initial prototype of focus depth

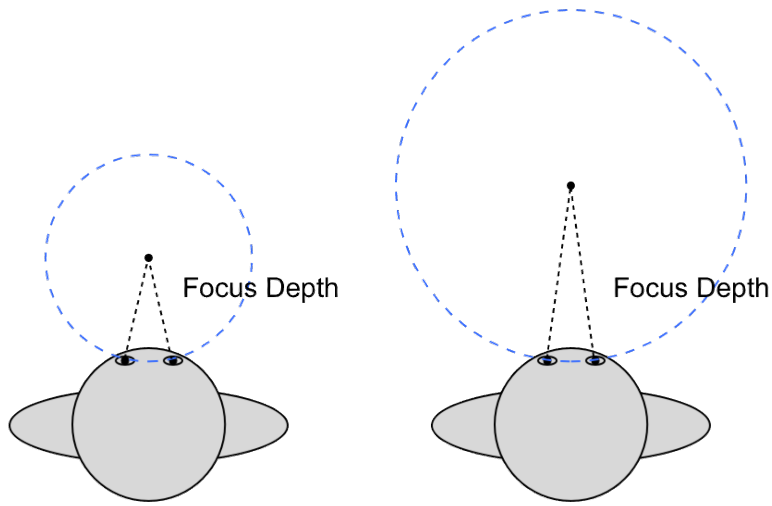

Figure 4.37: Focus Depth of the user where left is close and right is far

implementation in a standard VR headset, which also implies an integration with AR HMDs. Although a couple of researchers already implemented binocular eye tracking systems in VR [37], prior to this work we are not aware of any research using focus depth tracking as an interaction modality. Our contributions are as follows: (1) We implemented a custom prototype to use eye gaze focus depth as novel input modality for VR, (2) We show a sample application using focus as a switching mechanism and show that interactions using focus depth information are comparable to explicit input of a scroll wheel, and (3) we present guidelines and application cases for using eye gaze depth in VR systems.

Since the Pupil trackers do not support virtual environments by default, a custom plugin was written for the tracking software with Python that enables the raw data from the trackers to directly stream into Unity via open sound control (OSC). Unity then reads these values directly into the virtual environment. We are using the two normal vectors of the iris for both eyes (provided by the pupil software) to perform the depth calculation, detecting the intersection point or the vector that represents the shortest distance between the two vectors. The focus depth is accurate for distances between 5 to 25 meters in the virtual space. For the user study, we use a calibration system based on the K- nearest neighbor (KNN) algorithm that teaches the system to recognize two layers of depth in the virtual world. By selecting K= 3, KNN calculates the Euclidean distances of the

Figure 4.38: The experimental setup

current eye gaze with the trained values to determine the two closes values of K that contains the information of the layer currently being observed. This allows a robust user dependent recognition.

User Study

We conducted a case study to determine the usability of the proposed system by comparing two methods: scroll-based and gaze-based. The scroll-based method utilizes a more conventional approach where the user is required to obtain scores by touching a sphere, which is controlled with a mouse, to a 2D square placed in front of the user. However, the squares position also change in the z-axis (depth). The scroll wheel allows the sphere to move in the z-axis. For the gaze-based method, the scroll wheels function of depth control is substituted with eye gaze. After the calibration phase, the task is no different than the scroll-based method, except that the user is now able to control the spheres depth simply by focusing near or far. A total of 10 participants, consisting of 6 males and 4 females aged between 20 and 25 and have variable degree of eye sight clarity were given a score-based task in the VR environment. At the end of the session, each participant is then required to complete the System Usability Scale (SUS) score questionnaire to determine the gaze-based methods intuitiveness.

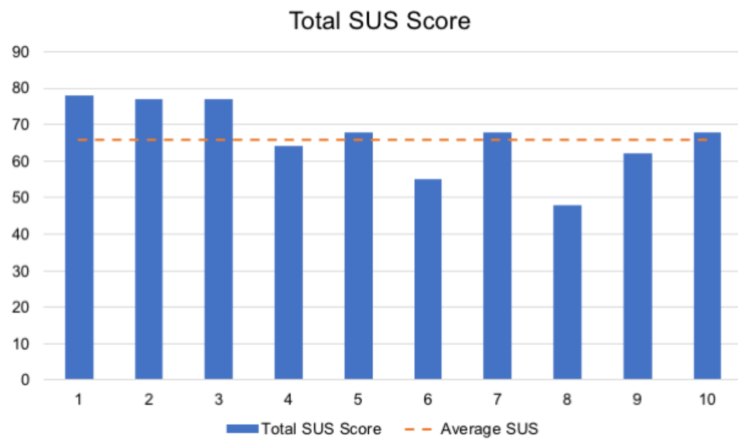

Figure 4.39: SUS Score of 65.5 of the proposed method

Results

The achieved scores are rather comparable overall with gaze-based method marginally higher in all three rounds. Two of the participants wore spectacles and had no previous experience with eye tracking and VR. One of the participant needed to remove it while wearing the HMD, whereas the other was able to fit the spectacles in it. Participant 6 and participant 8 suffers from both short sightedness, while participant 8 suffers from minor diplopia. This leads to difficulty in focusing at objects. If the results for both of these participants were excluded, the new SUS score would be 69, which is above the average SUS score that is deemed as a favorable system. Applying a T-Test on the score results show that for the first session, a p-value of 0.804 was obtained. For the following second and third session, the value steadily decreases from 0.4266 to 0.295 respectively. There is no statistical significance between the scroll-wheel and focus-depth based method. Indicating that our method is at least comparable to the scroll- wheel implementation.

Discussion and Limitation

Overall, we demonstrated an initial implementation of eye gaze input in VR.

Even though Transparent Reality provided us with an acceptable SUS score, some of the feedback from the participants were that prolong use could lead to eye

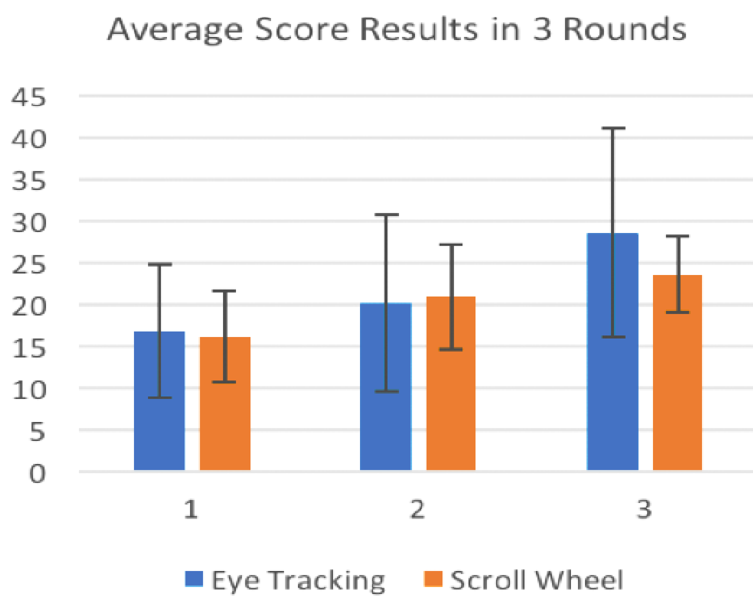

Figure 4.40: The average scores for both scroll-based and gaze-based method for the first, second and third trial

strain. Similar to real life, a constant shift in depth perception of the eyes can be quite strenuous. Therefore, it can only be used sparingly and is very situational. We would also like to touch on the implementation of KNN for classification, which we did not compare with with other machine learning solutions. Since deep learning has been gaining popularity as the definitive algorithm for machine learning, a comparison between deep learning with various machine learning methods could be made to improve the overall classification.

For this final approach, we found that one of the main drawbacks for using eye tracking for selection and activation is that it could quickly lead to eye fatigue, so care needs to be taken. However, the results show that is a feasible method to actually replace our selection and activation motions to that of only the eyes, because it triggers much faster and barely requires any physical motion.

4.6. Summary

To summarized all the approach in this work, the first step into understanding Convex Interactions is to recognize the parameters that were tweaked so that it can be reproduced by other researchers. In Chapter 3, it was established that the main five parameters used in the approaches are space type, interaction type, mapping type, motion type, and output type. In this section, I will explain what this means for Convex Interactions in two ways; the reason why we can perceive it as being more natural or intuitive than what we perform everyday in terms of cognition and neurons, as well as the generated human meta-model as the main output for this thesis, so that researchers from this field can reference as well as create their own form of Convex Interactions.

4.6.1 Hacking the Human Brain

In this section, I would like to discuss the reasoning behind the simple phenomena of how we as humans are able to adapt to a new interaction, and even briefly overwrite what we initially deem as more natural in favor of something that is more efficient. One of the earliest example given in this thesis is when humans first learn to walk. We started off from crawling, to walking with only two feet. This evolution in human locomotion is all due to the flexibility of the human brain to adapt to given situations and to develop methods of optimizing ourselves to not only understand, but also overcome our physical limitations. Evolution happens to all living beings on this planet, but what makes humans special mostly stems from our intelligence. Back to the example of human locomotion, we adapted to our environment by finding the most efficient way of moving around, and eventually learn that our hands are not needed to move around. Our brain gradually overwrites this information over time during our learning process. As a result, we just need our legs to move, though we still do swing our arms when walking. We know that walking with just 2 legs then, is simply more efficient, and in time, we grow to understand walking with 2 feet as the most natural way to move around. Therefore, for us to view Convex Interactions as being beyond this, we need to understand that it instead, is more efficient then walking with 2 legs.

This understanding process is called cognition. Cognition is defined as the mental process of acquiring knowledge and understanding. As Convex Interaction aims to overwrite what we are already used to performing on a daily basis, we need to change the information that is being fed. In this overwriting process, our cognition adapts and learns from the aforementioned information through the sensory receptors. When less space is used to move, then our cognition is reprogrammed to comprehend this. Of course, the learning process also highly depends on how we find the new input to be intuitive or natural, but because it is simply more efficient, this plays a large role in our learning process.

4.6.2 Neuroplasticity of the Peripersonal Neuron

Whenever a new locomotion method is developed and tested in VR, motion sickness tends to be the first issue that needs to be addressed. Even when out of VR, some people already experience motion sickness simply by just being in a driving car or or even watching a character walking around in a display in the first-person view. Why then, do methods like AnyOrbit can overall produce less motion sickness? This is because what we see does not correlate with out body motion. In a car, our bodies are not moving, yet our vision knows that the body is being moved, causing a sensory mismatch. This is a very good example to show how the human brain has been cheated or hacked, or has simply adapted to a situation when our visuals are aligned with our motion. AnyOrbit changes the output of the locomotion to be that of rotational, but motion sickness is less because the mapping of input is to the head that is being rotated to perform it. Since the motion is correlated to the output, motion sickness can therefore be avoided because there is a sensory alignment between motion and vision. Therefore, it is actually fine to create a non-linear output, as long as the input motion correlates with it, even if it is being mapped differently.

Regarding mapping, is has already been previously proven that humans can adapt to different limb mapping and a shuffle in body schema [129]. Humans are able to adapt to additional limbs, as well as the change of input mapping, therefore I also leverage this in an effort to reduce space consumption. There is a close relation between body schema with peripersonal space, because an effective control of the human body mainly depends on the integrated representation of the body (body schema) as well as the space around the body (peripersonal space).

In Chapter 1 and Chapter 2, I touched on the existence of peripersonal neurons, which are clusters of neurons that exist in the human brain with the purpose of encoding our peripersonal space and spatial sense. It provides humans with a mental model, or a meta-model of the space around us to the point where, even with our eyes closed, we know where the positions of our limbs are. Another term closely related to this is proprioception. Proprioception is defined as the awareness of one's own body, and is detected from the nerves in the body as well as the canals in the inner ear. Because of this, it is also subjected to neuroplasticity, giving it the ability to keep changing depending on the information from our senses. The brain's processing is overall more complex when there is activity closer to the peripersonal space as it actually involves more sensory modalities [55]. Based on what we found from this work, interactions that are closer to the peripersonal space greatly effects our cognition and ability to adapt to it faster due to this additional sensory information.

Furthermore, because of the peripersonal neurons and its connections with these nerves, humans have a general meta-data of the space around them. We do not care too much about the specifics; the angle of rotation or the amount of displacement. In Unconstrained Neck [137], participants are able to adapt to the change in vision where the neck rotation results to a bigger change of view. What matters more is the correlated motion with the vision, as explained earlier. This study now shows that there are plenty more factors and parameters that can be tweaked, evident by each of the approaches used, and that space greatly effects each of them.

Figure 4.41 shows the relation between the peripersonal plasticity of the human space with the neuroplasticity of the brain found in this study. Interactions closer to our core, or body, allows us to adapt faster to that motion. In the brain, the regions that are in charge of the cognition process and the spatial information are the frontal lobe and the parietal lobe respectively. These two regions cover most of the brain, though this work triggers specifically the motor cortex in the frontal lobe and the somatosensory strip in the parietal lobe. The motor cortex reads the information from the motions of the limbs whereas the somatosensory strip obtains information from the sensory receptors, which in turn influences the peripersonal neurons.

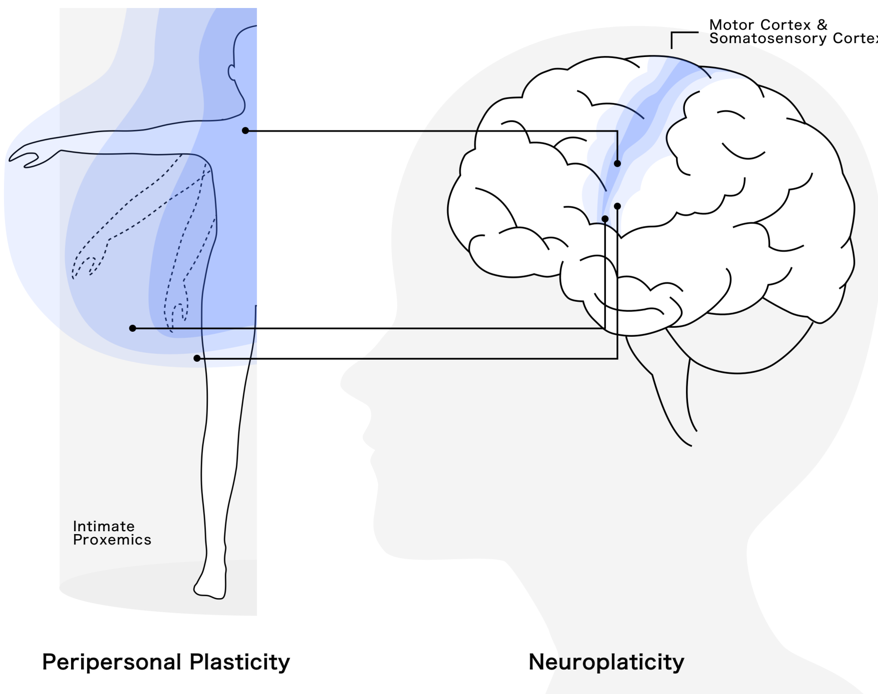

Figure 4.41: Correlation between peripersonal plasticity and neuroplasticity, where interactions closer to the body can be more efficient and intuitive quicker

So from this, we understand that the results obtained from the approaches mentioned in this thesis mainly stems from the neuroplasticity of the human brain. However, this information alone cannot benefit engineers/designers/researchers to further push the boundaries of interaction design if the reason is simply attributed to neurons.

4.6.3 Human Meta-Model for Convex Interaction

The concept of space is itself, one of the main parameters that has been established in this work. In Chapter 3 though, I further established several other parameters that have been manipulated in the approach, as a result that started from spatial manipulation. These parameters are the space type, interaction type, mapping type, motion type, and output type. Based on these parameters, engineers/designers/researches can design and develop their own form of Convex Interactions based on these parameters. Previously, I also mentioned that humans have a mental model of the space around us, and the core contribution from this thesis is understanding the parameters from this model that we can extract to create efficient interactions. Figure 4.42 shows the meta-model, which is a model based on that human mental model.

It can be seen that each of the developed approach are from a different combinations of the parameters. For example, Transparent Reality is an interaction mechanic developed within the peripersonal space (using only eye movement), based on a serial-type interaction (selection and activation which is a series of movements to achieve the desired output), rearranges the mapping type (having the first input of eye motion being coupled with the final output of picking an object), a different motion type (changing the depth of focus of the eye as opposed to using arm or finger movement), and a non-linear output (makes the object's position change it's position in terms of distance from the user, or depth). Therefore, other engineers/designers/researchers may use these various parameters to either recreate the Convex Interactions developed in this thesis, or create a new form of interaction by combining these parameters into something new. However, further tests would then need to be conducted to understand the user's perception towards the said new interaction method.

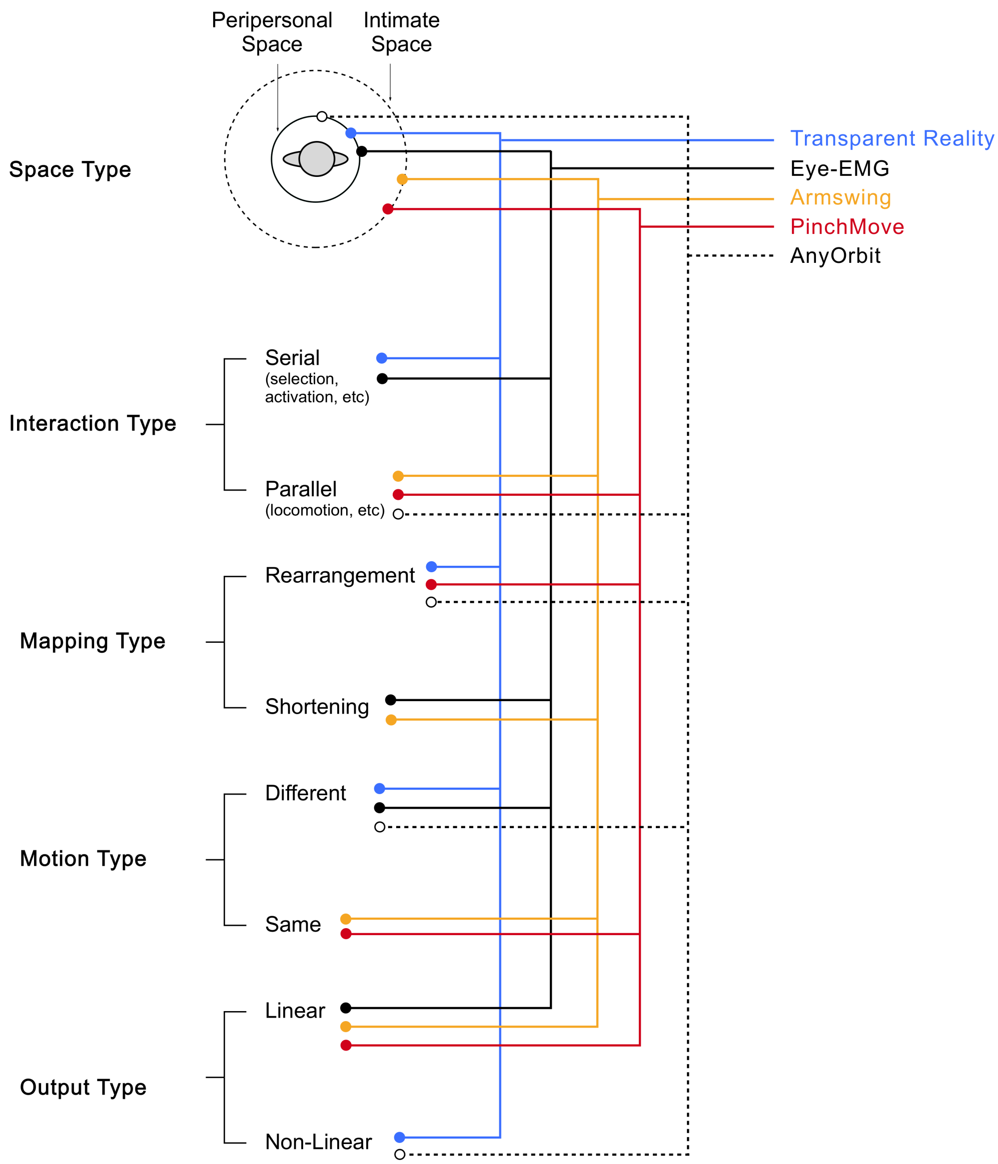

Figure 4.42: The human meta-model for Convex Interactions

Chapter 5

Application Scenarios

This chapter explores the potential use cases and scenarios that can benefit from Convex Interactions. The first section is the use of Convex Interactions that can potentially promote VR to be more ubiquitous, serving well especially in collaborative environments and being able to open new VR interaction spaces in the future. The second section focuses more on the bigger picture on how Convex Interactions can push mankind towards a more efficient interaction method as we move towards a machine-oriented future.

5.1. Applications in VR

These scenarios depends on two factors; space or proxemics consideration, and collaborative or social interaction, illustrated in Figure 5.1. Certain scenarios contain more users around us, whereas other scenarios move towards the degree of collaboration or social interaction. For example, one of our recent work, CleaVR, delved into the development of a collaborative environment for interior design and was presented as a poster paper at Siggraph 2017 [102]. This paper was co-authored by myself, Benjamin Outram, Benjamin Tag, Megumi Isogai, Daisuke Ochi, Hideaki Kimata and Kai Kunze. This work showed the potential of a collaborative social space virtually, with a high degree of collaboration but low proxemics consideration since it featured remote collaboration. Among the other application scenarios that are covered here are spectating sports in a stadium, watching sports at home, having a meeting, and performing collaborative art in

VR.

Current methods for spectating live sports events has not changed over the years. In the future though, the possibility of mixed-reality adoption is possible. Sports spectating generally involves hundreds of people tightly packed in a stadium to spectate their favorite sports. This leads to actual proxemics concern, even though the collaborative nature is minimum.

The next scenario with minimum collaboration as well as minimal space used is watching sports in the comfort of the living room. Such a situation would involve roughly three to four other users (family members and friends), where some social interaction takes place though without much worry on intrusion.

For a scenario that requires more collaboration in a relatively smaller space, an appropriate scenario would be a meeting, group discussion, or group revision between several professionals like engineers or product designers. Social interaction would be necessary, with some care regarding intrusion of personal space.

The last scenario to be considered is a major collaborative art project. Such a scenario may involve several artists working together on a large virtual canvas to create a piece of VR art. In this scenario, care needs to be taken for intrusiveness especially, as collision will effect the paintings of the other artists. Another scenario that falls into this category would be a massive local VR multiplayer game, which can be both collaborative or competitive.

Generally though, all scenarios share the same social space and are variations of Figure 2.1 depending on the aforementioned two factors. Some of these scenarios already exist in today's current applications, whereas some of them are projected futures that could very well be adopted in the next 5 to 10 years. In each of these scenarios, the presence of other users are shown to very likely be within the personal space of each other, thus making non-intrusive interactions a useful alternative to interacting with virtual content, or with each other virtually.

Next, I delve deeper into each proposed convex interactions and discuss the possible scenarios and applications where each of them can benefit most from.

5.1.1 Arm Swing

Arm swing was an interaction proposed to convex social proxemic space to personal, thus its application scenarios can be rather unique. The developed system

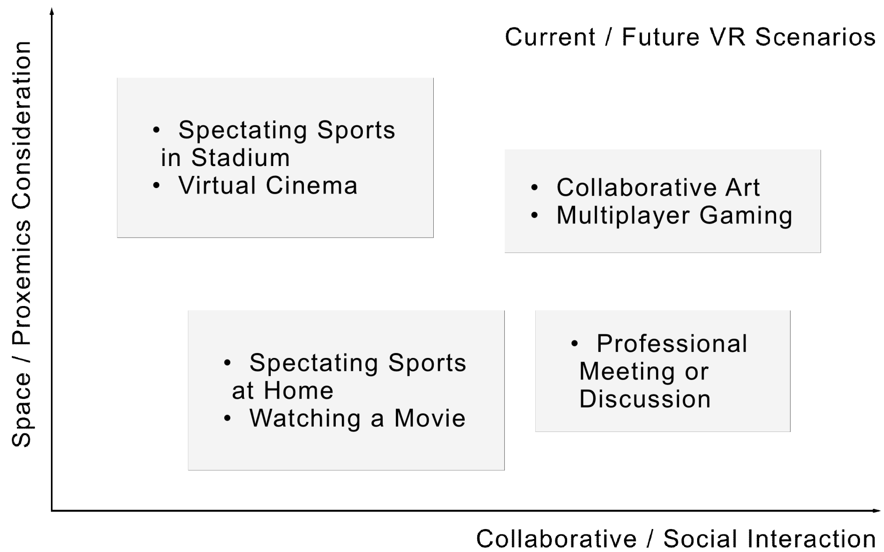

Figure 5.1: Categorization of each application scenarios depending on space and social interaction

was catered towards standing use, though it can still work for sitting purposes since the user only requires arm movement. Since it was developed to balance energy consumption with immersion, it can firstly be used for collaborative or competitive gaming scenarios; picture a multiplayer AR or VR race achieved entirely with arm swinging. Next, since arm swinging was compared with WIP solutions, thus any scenarios where WIP can be used also caters to arm swinging, such as space exploration use cases (interior design, virtual tours, etc.). Furthermore, its use case can be expanded to a wider range of audience; an example being the elderly or even disabled people who wish to experience natural VR interactions. This also opens up further design considerations for this category of audience, though we shall relegate this to future works.

5.1.2 PinchMove

PinchMove was designed with a more specific use case, with accuracy and precision being the main factor as opposed to immersion. For accuracy, it can be said almost all other existing technology has an input method that is deemed to be most accurate depending on application. Gaming can be seen as a good example. Even though a desktop computer supports many kinds of controllers, the mouse is overall the most accurate in selection, evident by how first person shooter (FPS) games are generally more competitive for the desktop computer [59]. For games using the third-person camera, a gamepad becomes ideal when compared to a mouse. Racing games instead favor an actual steering wheel controller. These input methods are the most accurate for each of these types of games, and can also be reflected in professional use. For example, we generally prefer to use a mouse for software tools like CAD modeling or video editing. For AR/VR, since it was developed to mimic real life, naturally movement is most accurate when we move the same way we do in real life. However, clearly this is not a feasible method since physical space is limited, which brings us back to the issue of proxemics. Teleporting, another navigation method that is currently the definitive navigation mechanic for VR, on the other hand can be jarring and disorienting.

As previously mentioned, PinchMove was designed for specific scenarios that are near-field (personal proxemic) and prioritizes accuracy. An initially discussed scenario was to apply PinchMove in collaborative social spaces like an office environment, where space is limited, objects are around the near vicinity, and accuracy is needed. In a VR office conference scenario, each participant can easily navigate to the front of a desk, or towards a white board accurately. One-hand navigation may be used if the other hand is occupied with a marker pen. A similar scenario would be a workshop simulation for technicians. Another suitable proposed scenario would be for a simulation of a space ship for astronauts. Since PinchMove relies on physically pinching or grabbing the environment by hand, this mechanic is suitable for zero gravity where an astronaut actually needs to physically grab the inner walls of a space ship to navigate. The rotation mechanic allows them to accurately position their orientation, which is also suitable for the ergonomic design of the interior of a space ship. PinchMove simply needs to be modified to allow navigation through all axes, as opposed to just a plane in our user study. Similar scenarios would be underwater navigation and climbing. A participant from our user study suggested that this method works well with a surgery training simulator where accurate control of the camera is necessary to observe vital organs closely. Another participant mentioned that PinchMove works well when there are plenty of objects surrounding the user to be used as an anchor point for movement. Such a scenario would be for logistic managements and storing, where a virtual forklift can be controlled in this manner. Another suggested use case is for sharing empathy or experiencing disability, particularly using the one-handed navigation which forces the user to move around only with one hand. For gaming, it can lead to interesting mechanics, such as playing a baby that crawls around. Finally, a participant suggested that modeling or painting applications like computer-aided design (CAD) software or Tiltbrush can benefit from our method due to the accuracy which is vital when creating 3D content. Since PinchMove allows full navigation without physically moving at all (except the arms), it is even possible for users to be immersed in any of these suggested scenarios where physical space may be extremely limiting and interactions need to be kept at a personal proxemics distance, such as on a train or plane ride. It provides the required navigation feature without sacrificing accuracy.

5.1.3 AnyOrbit

AnyOrbit looks into the next step of convex interactions, which is convexing into intimate space using only head movement. However, it is more on a unique take towards navigation and cannot outright replace conventional mechanics. Since navigation is tied to head rotation, this will effect the content that is being consumed as well. Therefore, it is specifically used for both a combination of navigation and media consumption.

Although the exocentric rotation in AnyOrbit is unusual, movement is linked with our head rotation, which we showed mitigates simulator sickness, and may also be beneficial for immersion and presence. The technique potentially leads to new types of interactive media experience, and can be applied widely to sports and e-Sports spectating, 3D recorded media, data visualization and games. It can also be used for spectating sports, giving each viewer a choice of either using the directed mode or free-form mode (camera control with mouse or eye gaze). Sports is also a good proxemics example. Spectating it can either be in a living room with close friends within the social proxemic space, or at a stadium with other viewers being within an intimate proxemic space. Since AnyOrbit can be used within the intimate space, it caters to both scenarios.

5.1.4 GazeSphere

GazeSphere is a variation of AnyOrbit that combines with eye gaze and is tailored specifically for navigating 360-degree-video environments. We asked the question during the development phase of GazeSphere; how do we navigate the real world with minimal effort? Therfore, GazeSphere provides users with a method akin to Google Street View for VR; a method to select a path of navigation and navigate to that selected point, all the while being hands free. This form of interaction and navigation makes it ideal for AR/VR story telling, where the viewer may freely orbit around points of interest, that move on the next plot point depending on the choice selected using eye gaze. This also promotes the application in collaborative gaming, where multiple players may support features like branching storylines or puzzle games. For professional applications like art spectating, each user may select interesting points via eye gaze, and navigate there using head rotation as well to allow interesting shifts in perspectives. However, this is arguable not suitable for CAD modeling, which favors more precise navigation.

5.1.5 Muscle with Eye Gaze

We designed several applications that can fully benefit from the proposed interaction. Since AR/VR is a diverse platform, the applications are divided into their respective fields for interior and engineering design, entertainment and gaming, as well as UI interface selection and media consumption. For each of these applications, we show how the user can easily select and activate elements that are present depending on the application that is hands-free, time saving, and unobtrusive. These applications serve as a proof of concept on how eye gaze with EMG can provide a unique alternative to interaction, whether it be gaming on the go or having a virtual business conference. Each of this applications will be explained through its categorization for both the target acquisition phase and target action phase. We explored four application use cases based on our novel input modality: 1) interior design explorations, 2) gaming, and 3) text input.

One of the core benefits when it comes to VR is the ability to place users in a virtual environment as though they are actually there. For AR, it further augments the environment with additional virtual content. For professional use,

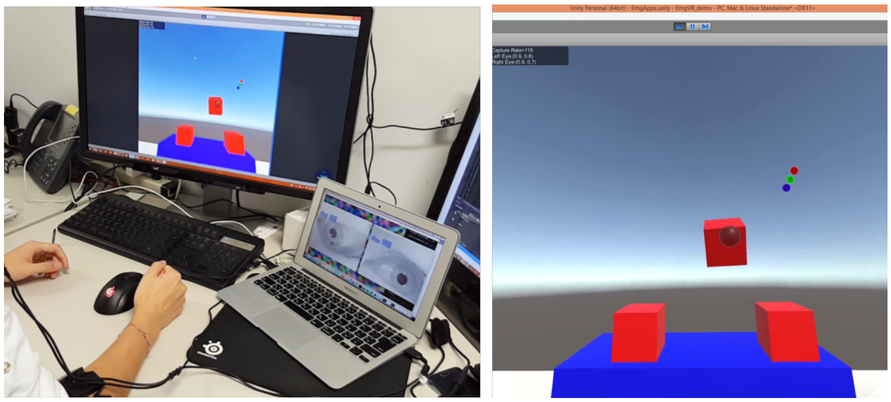

Figure 5.2: **Pick-and-place task**

one such usage would be assessing an interior design. In such an environment, minimalistic UI is best so that it will not obstruct the designer's view and to allow careful observation and assessment of the environment. In our implemented prototype, navigation and pick-and-place is provided through a discrete action of the target action phase. If the user wishes to navigate to a position, they can simply look at the ground of that point and perform a muscle contraction to teleport to that location. Teleportation navigation is increasingly more popular as it negates the effect of motion sickness [27]. We also implemented a pick-and-place tool shown in Fig 5.2, allowing the user to look at a particular object of interest, pick it up via short muscle contraction, and place it at any designated spot by performing another muscle contraction. To summarize, this particular application is a pick-and-place scenario by moving furniture while navigating interior spaces.

This generation of AR/VR products focus on gaming, therefore it is only natural to consider some kind of gaming function. We created a simple shooting scene shown in Fig 5.3 where the user views the world in a first person view with the gun placed at the lower right corner, similar to most first person shooter (FPS) games. We equip the user with a pistol and allow the user to toggle between full-auto and semi-auto firing. Full-auto allows the user to continuously fire the rifle for as long as their muscle contracts, while semi-auto fires a single bullet per muscle contraction. This effectively switches between both kinds of target action phase. One of the interesting benefits of using physiological signals is that in this use case, contraction of the arm muscle could simulate the recoil of a gun, whereas

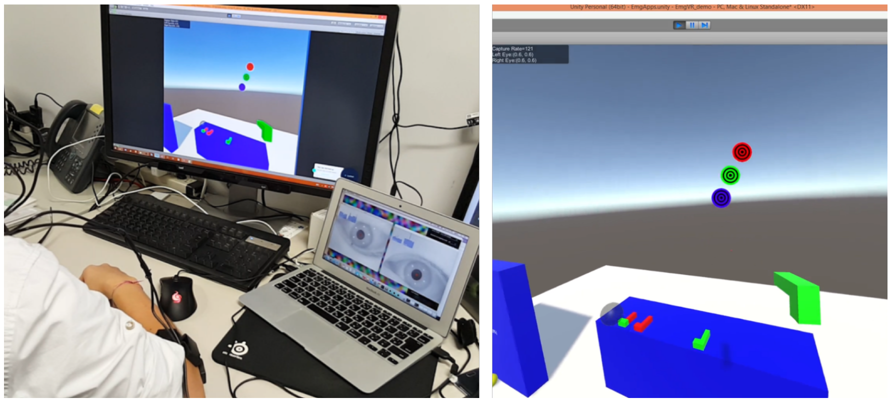

Figure 5.3: **Firing a gun in full-auto and semi-auto**

aiming with the eyes simply feels natural, leading to a greater level of immersion in gaming though further studies are required to test this hypothesis.

We also created a generalized UI system to demonstrate the feasibility of the proposed interaction mechanic. The first UI is a number pad illustrated in Fig 5.4 that allows the user to simply look at a number and contract their muscle to select it. The second prototype is a series of knobs and graphs. Each of this interface depends on the user's amount of muscle contraction. Maximum contraction will maximize the knobs and graphs, while relaxing the muscle reduces it back to 0. These UIs server as a proof of concept for them to be applied in various other usages like menu selection, scrolling, and media control without the presence of physical buttons or other devices that occupies the users' hands. Especially in a collaborative environment, these tools can be used for productivity purposes.

These example applications show the diverse feasibility of the proposed multimodal convex interaction, particularly because it uses minimal space. If the proposed scenario uses wide gestures instead, then applications like gaming or productivity tasks would be difficult to achieve collaboratively.

5.1.6 Transparent Reality

Transparent Reality, which as interaction mechanic that uses focus depth for selection, will be further explored here. Since it offers a new layer of interaction in AR/VR, it has the potential to also provide a hands-free experience that preserves the immersion. For example, this method of selection would be useful for heads-

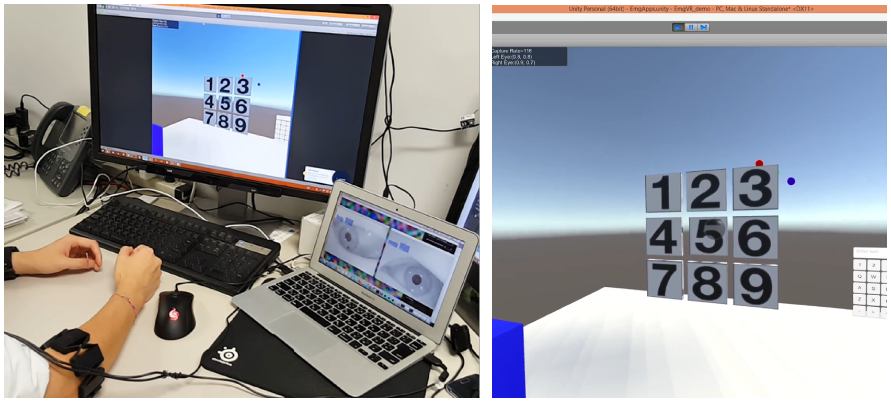

Figure 5.4: **Number pad interaction**

Figure 5.5: Transparent Reality system that uses depth of eye gaze to visualize HUD

up-display (HUD) based interaction where the user just focuses close to see the HUD, and focuses far to see the main content. This is particularly useful in spectating sports, as shown in Figure 5.5.

Along the lines of the transparent HUD, focus depth can also be used for menu selection tasks, where different focus depths are associated with different menus. Another application in AR/VR is that it can provide a user with a window to the physical world by mounting a camera on the HMD. By focusing close, the user may switch back to the physical world, while focusing far causes the physical world layer to fade away.

I also present a novel technique of foveated rendering to keep the computing workload low and create a more natural image that is clear in the focused field, but blurred outside that field. This work is adapted from a poster paper presented at Siggraph 2016 [104]. The paper was co-authored by myself, Benjamin Tag, Benjamin Outram, Noriyasu Vontin, Kazunori Sugiura and Kai Kunze. The proposed

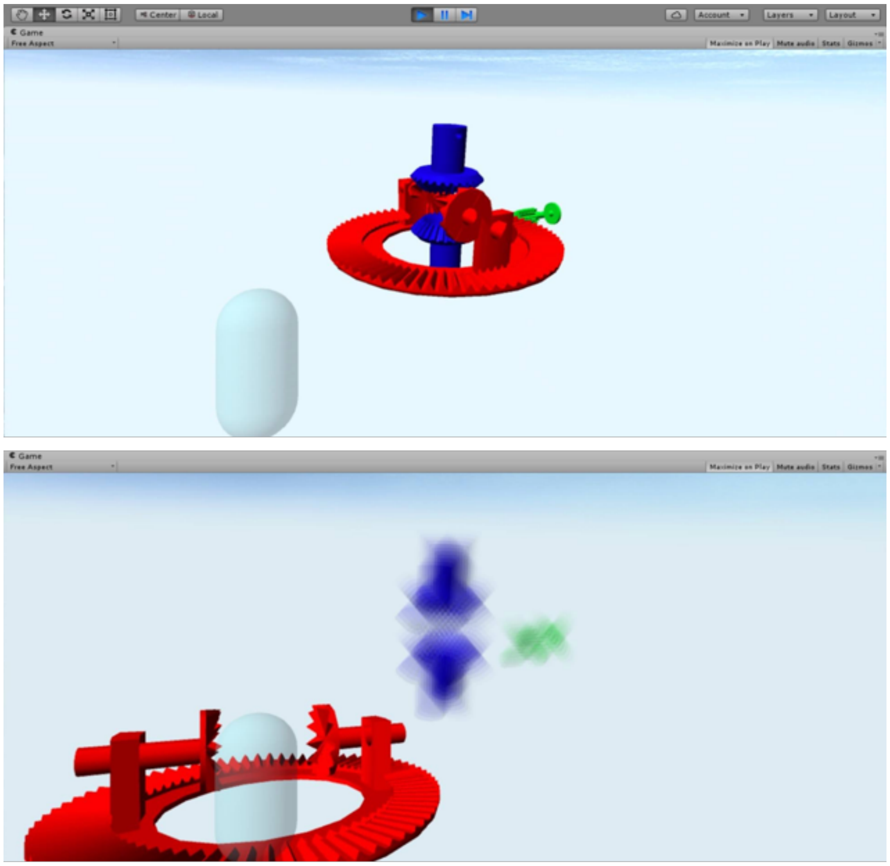

Figure 5.6:) Example Application: Gear Assembly foveated rendering depending on depth of focus of the eye

GazeSim system uses the depth recognition from Transparent Reality to achieve a more true-to-life foveal rendering. The human eyes adjusts its depth of focus so that when we focus on a close object, the background becomes blurry. GazeSim achieves a similar effect for VR where the user's depth of eye gaze changes the rendering clarity. Compared to other foveal rendering solutions that merely uses eye tracking to render a specific portion of the scene to be clear [109], our solution is a more accurate representation of the human eye. These interaction mechanics allow VR to be used more subtly with its hands-free nature.

5.2. Future of Human Interaction

I thoroughly explained how Convex Interactions can benefit VR itself in the previous section. Here, I will explain about how Convex Interactions aim to benefit as mankind's next step of interaction evolution.

It is clear that, interaction such as ArmSwing or Eye-EMG clearly cannot be achieved by humans alone; we have anatomical limitations to how we can move regardless of how much we can hack our cognition. However, what I found from this work is that, there are motions and gestures that can be performed where we perceive them to be natural and more efficient. To realize this in real life scenarios of actual locomotion or pick-and-place, there exists two possibilities; one where prosthesis can be used to not only assist the disabled, but retrain them to be more efficient in their daily lives. Current design of artificial limbs, or prosthesis, for the disabled are meant to directly replace the missing limb, without changing the actual motion. However, this work proposes that prosthesis can now be designed based on Convex Interactions to change how the disabled interact with the environment for the better.

Now, I would like to look into the possibility of beyond natural interactions towards the future of interaction itself. For those more physically fit, Convex Interactions envisions a future where man is married with machine more. Designing machines that can augment human's movement in space can largely benefit from the findings in this work. Looking at current human augmentation technologies, we can see that the design, like prosthesis, are also rarely about redefining the human motion. If we look at these technologies becoming mainstream in the next several decades, then Convex Interactions will play a very important role in our evolution, because it creates a basis of understanding on how we can fundamentally change our way of motion and input. For example, the images below show us the possible designs that have been proposed for human augmentation machines. The idea of not just improving the performance of an existing limb, but also the addition of limbs will open new paradigms when coupled with Convex Interactions.

Chapter 6

Conclusion and Future Works

In this thesis, I investigated the idea of an interaction method that exists beyond what we claim to be natural by being more efficient. Through a series of approach and investigation with various parameters stemming from proxemics and peripersonal space, I explored how Convex Interactions can possibly achieve this. In the following section, we provide an overall summary of this research, look into the limitations of said method, ans conclude with the future directions.

6.1. Summary

Interactions with digital space depends on the hardware and the input device presented with it. The mouse and keyboard was and still is how we interact with computers, whereas for smartphones, the screen itself becomes the input modality. For us humans, it is the motions from our limbs, and the physiological signals that trigger them. From these inputs methods, they are the definitive ways of interacting with their particular digital space, for one particular reason; their efficiency. Therefore, this work stresses on that key point, of finding what are the parameters that can create a more efficient way of everyday interaction.

In the digital space, we can look at VR to be a tool or platform that is closest to that of the real world in terms of interaction with it. Due to its ability to track the user's head and hands position in space, it gives the freedom to researchers to use it to further develop new gesture-based modalities while understanding its effect on the human's perception. Convex Interactions is such a modality. By leveraging the concept of space around the human and its connection with our cognition and neuroplasticity, it has proven that fundamental interactions that are an improvement over what we do everyday can still exist in a better form in terms of efficiency. From space, further parameters were then defined; type of interaction, type of mapping, type of motion and finally type of output. All these parameters can be tweaked to understand and even develop new forms of motion that are not only just efficient, but leads to it being more natural, intuitive and simply better than how we interact today. In understanding this, we also know the flexibility of the human understanding, how the concept of close proximity space plays a large role in this, and how we can eventually use this to create micro interactions as an extension of the body schema for improving present VR ubiquity, and future human interactions.

6.2. Limitations

As with any research, the limitations that exists in this study needs to be clarified and discussed. Arm swing, PinchMove, AnyOrbit, GazeSphere, eye-EMG, Transparent Reality, and GazeSim were all the introduced convex interactions each with their own set of limitations previously discussed in Chapter 4. This section explores the limitation that exist with the concept of convex interaction in general, encompassing all the proposed interactions and how they can be further improved in the future.

The primary limitation currently is the number of input and sensing methods that was explored. There exists plenty of physiological signals that can be detected for each person, and therefore this work only focuses on signals that are more explicit than implicit. In other words, we focus on signals that can be controlled by the user to some degree, instead of signals that are generally more useful as feedback, such as heart rate. Furthermore, the selected sensing methods can be achieved through HMD modifications or using the existing peripheral, which is a major advantage over introducing additional sensing wearables, with a key difference being the the Myo sensor for sensing forearm muscle contraction. Since one of the key strength of physiological signals is the implicit sensing, we can further improve convex interactions in the future by using both explicit and

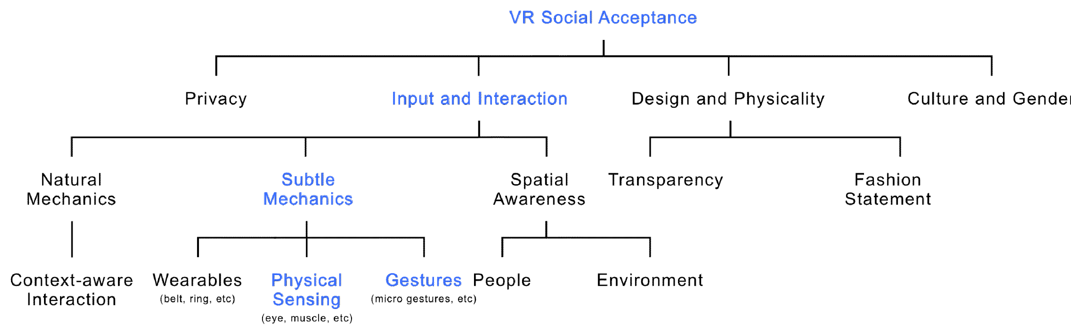

Figure 6.1: Expanded chart for social acceptance of the next generation VR technology

implicit forms of interactions.

Next, I would simply like to touch on the notion that Convex Interactions are aimed for a future when man marries machine more intimately for a more augmented human scenario. Clearly, this is just the first step into understanding the correct and effective motions. To fully realise this scenario, we also need to approach it from the hardware perspective, by actually developing physical tools, proshesis, or additional limbs that can actually put these interaction methods to use in the physical space. As of this moment, they are merely a simulation in the virtual world. This leads to the future works direction, where I discuss how to approach this next.

Finally, I would like to stress on the social acceptance direction for Convex Interactions, based on its proposed application scenarios. In the previous chapter, I suggested how this work can contribute towards making VR more ubiquitous in the shorter term. For this, there are still many obstacles to overcome besides redefining the input space, which is one possible direction. The idea of VR being anytime, anywhere is an alluring prospect from the research side, but several other key issues may arise as well. One of them being the social acceptance of convex interactions. These input methods may be usable and advantageous, but when used in the wild, may cause head to turns. Imagine someone pinching midair, or constantly moving his or her head around while standing in a bus. This will undoubtedly seem awkward to other nearby people. Therefore, social acceptance becomes a key research question when introducing convex interactions to the public.

Figure 6.1 shows us a glimpse into the future expanded research areas for this field, and where convex interactions shown in the red boxes, only play a smaller portion in the overall big picture. All these issues need to be addressed so that we may open the door for the next generation of ubiquitous technology.

For the next main application, where Convex Interactions is for redefining human interaction, this also introduces further social acceptability questions. Picture a point where, everyone only perceives that they can move around space simply by swinging their arms. This makes their legs become relatively useless, which will then possibly lead to deterioration of limb performance. Our ability to locomote may become more efficient when a machine can move us around by reading our arm swinging motion, but if this is at the sacrifice of our legs gradually, this may not bode well for human evolution. Of course, this is assuming that such an interaction method becomes mainstream for many years, until the generation of "Convex Interactions Native" might possibly not understand how our legs function. To overcome this, we need to look further into the social acceptance of this work.

6.3. Future Works and Final Remarks

As previously mentioned, the obvious next step would be to bring what has been discovered in this work into the physical space. Since Convex Interactions is first coined in this work, the first step was using VR as the main tool to develop it. The next step in physical space needs to move to automation and robotics to develop physical prototypes.

Another future work direction I would like to touch on is a quantitative approach to understanding natural and/or intuitive interactions. Natural interaction still remains a continuous field of research in the HCI community because there is a need for standardizing what is considered natural [4, 94]. It is agreed upon that gestures promote natural interactions, however, with the lack of standardization in HCI, natural interactions do not necessarily mean a better form of interaction. Therefore, one of the possible directions for Convex Interactions to gain traction is to perform a empirical standardization study of the various interaction mechanics that exist for HCI, AR/VR particularly, which is ever growing, so that we can

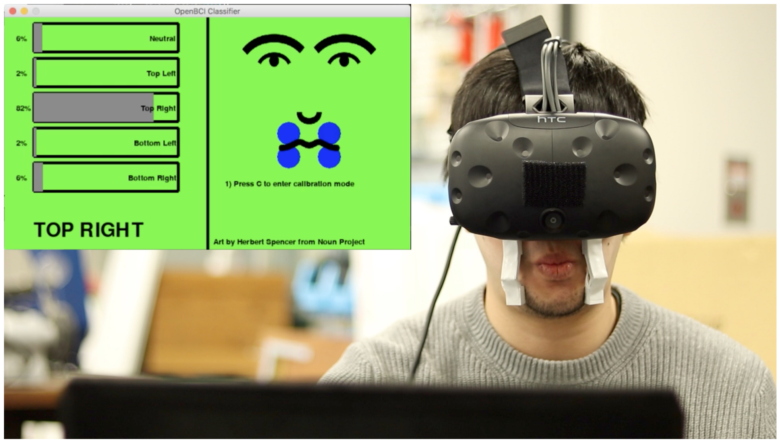

Figure 6.2: Make-a-Face, an interaction mechanic using tongue gestures

quantitatively define what is natural for each user. With gestures closely related to natural interactions, several works are still ongoing to properly classify and standardize them, such as a model based approach for different sensors used and it's corresponding gesture [142].

One additional sensing method that is planned for the pipeline of Convex Interactions is the sensing of facial muscles. Make-a-Face, shown in Figure 6.2 is a system that mounts EMG sensors on the face to detect tongue movement and facial deformation as input by classifying gestures using the Random Forest algorithm. However, for this to be deemed as a natural interaction, we are currently planning potential scenarios where tongue-based gestures are normal.

We are also looking into an easily accessible facial capture system that can be used to interact with virtual environments in a more implicit manner. Facefy shown in Figure is a system that uses facial expressions to dynamically change the content that is being viewed. It uses the Iphone X's front facing True Depth camera for uncalibrated facial tracking. However, one clear downside is that the system cannot be used with a HMD since it requires a clear view of the user's face, thus it was developed for screen-based AR. Nevertheless, this is just the first step of coupling Convex Interaction with an emerging technology in mobile devices. Though very much at the early development stage, we have been working on both Make-a-Face and Facefy as an interaction mechanic in AR/VR with collaboration with NTT Media Intelligence Laboratories.

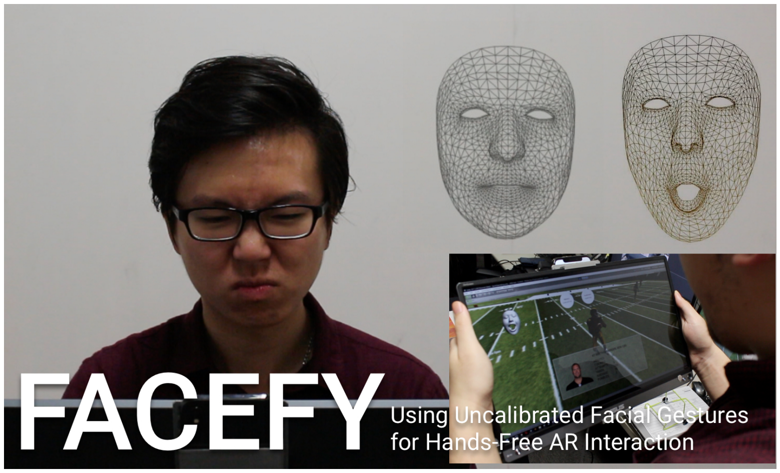

Figure 6.3: Facefy, a facial-based implicit interaction mechanic for AR environments

Looking back at Figure ??, we can see that there are still several rooms for others to develop and further improve from. For example, one could approach Convex Interactions by aiming for the intimate space, serial-type interactions, shortening mapping, different type of motion, and a non-linear output. The possibilities of combination for each of these parameters are vast, and can be possible future directions in exploring other forms of Convex Interactions.

Nowadays, the range of digital spaces are growing more steadily than physical space; new devices are always introduced that completely changes how we interact with them, but changes in the physical space has so far, not redefined our physical interactions. The ability for us to use what we learn in the digital space to adapt to the physical space is therefore, a very useful method to envision future interactions. VR is also a digital space that is and will continuously evolve in terms of hardware and software, where new sensing methods will be introduced in the future to the masses. The foreseeable future of VR being used anytime, anywhere is certainly not impossible given how fast the technology has been growing recently. The further future of human augmentation and the changes in our natural interactions though, may seem impossible to some, but a projected future always seem impossible until it becomes a reality.

References

[1] Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 11: Guidance on usability. *ISO 9241-11:1998* (1998), 22.

[2] Dictionary.com unabridged.

[3] Alkan, S., and Cagiltay, K. Studying computer game learning experience through eye tracking. *British Journal of Educational Technology 38*, 3 (2007), 538–542.

[4] Altakrouri, B., Gröschner, J., and Schrader, A. Documenting natural interactions. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, ACM (New York, NY, USA, 2013), 1173–1178.

[5] Anderson, Frank C., P. M., Anderson Frank C., P. M., Anderson, Frank C., P. M., and Anderson Frank C., P. M. Dynamic Optimization of Human Walking. *Journal of Biomechanical Engineering 123*, October (2001), 381–390.

[6] Ashbrook, D. L. *Enabling mobile microinteractions.* Georgia Institute of Technology, 2010.

[7] Bailey, T. Virtual Reality and the Body: Best Practices for Locomotion System Design.

[8] Bark, K., Tran, C., Fujimura, K., and Ng-Thow-Hing, V. Personal Navi: Benefits of an Augmented Reality Navigational Aid Using a See-Thru 3D Volumetric HUD. In *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '14, ACM (New York, NY, USA, 2014), 1:1–1:8.

[9] Barry, D. T., Gordon, K. E., and Hinton, G. G. Acoustic and surface EMG diagnosis of pediatric muscle disease. *Muscle & nerve 13*, 4 (1990), 286–290.

[10] Benford, S., and Fahlén, L. A spatial model of interaction in large virtual environments. In *Proceedings of the Third European Conference on Computer-Supported Cooperative Work 13–17 September 1993, Milan, Italy ECSCW93*, Springer (1993), 109–124.

[11] Benko, H., Saponas, T. S., Morris, D., and Tan, D. Enhancing input on and above the interactive surface with muscle sensing. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '09* (2009), 93.

[12] Benyon, D. Spaces of interaction, places for experience. *Synthesis Lectures on Human-Centered Information 7*, 2 (2014), 1–129.

[13] Bianchi-Berthouze, N., Kim, W. W., and Patel, D. Does body movement engage you more in digital game play? and why? In *Proceedings of the 2Nd International Conference on Affective Computing and Intelligent Interaction*, ACII '07, Springer-Verlag (Berlin, Heidelberg, 2007), 102–113.

[14] Bowman, D., Kruijff, E., LaViola Jr, J. J., and Poupyrev, I. P. *3D User Interfaces: Theory and Practice, CourseSmart eTextbook*. Addison-Wesley, 2004.

[15] Buchmann, V., Violich, S., Billinghurst, M., and Cockburn, A. FingARtips: gesture based direct manipulation in Augmented Reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, ACM (2004), 212–221.

[16] Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., and Kurtenbach, G. Stylecam: interactive stylized 3d navigation using integrated spatial & temporal controls. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, ACM (2002), 101–110.

[17] Cardoso, J. Comparison of gesture, gamepad, and gaze-based locomotion for VR worlds. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, ACM (2016), 319–320.

[18] Carlsson, C., and Hagsand, O. DIVE A multi-user virtual reality system. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, IEEE (1993), 394–400.

[19] Chatain, J., Demangeat, M., Brock, A. M., Laval, D., and Hachet, M. Exploring Input Modalities for Interacting with Augmented Paper Maps. In *Proceedings of the 27th Conference on L'Interaction Homme-Machine*, IHM '15, ACM (New York, NY, USA, 2015), 22:1–22:6.

[20] Chatterjee, I., Xiao, R., and Harrison, C. Gaze + Gesture : Expressive , Precise and Targeted Free-Space Interactions. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, C (2015), 131–138.

[21] Chen, W., Chen, J., and So, R. H. Y. Visually induced motion sickness: effects of translational visual motion along different axes. *Contemporary Ergonomics and Human Factors* (2011), 281–287.

[22] Chen, W., Plancoulaine, A., Férey, N., Touraine, D., Nelson, J., and Bourdot, P. 6dof navigation in virtual worlds: comparison of joystick-based and head-controlled paradigms. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, ACM (2013), 111–114.

[23] Chin, C. A., Barreto, A., Cremades, J. G., and Adjouadi, M. Integrated electromyogram and eye-gaze tracking cursor control system for computer users with motor disabilities. *Journal of rehabilitation research and development 45*, 1 (2008), 161.

[24] Choi, S.-H., Kim, H.-J., Hwang, S.-W., and Lee, J.-Y. Natural interaction for media consumption in vr environment. In *SIGGRAPH Asia 2017 Posters*, SA '17, ACM (New York, NY, USA, 2017), 26:1–26:2.

[25] Chung, J. C.-M., and Chung, J. C. Intuitive navigation in the targeting of radiation therapy treatment beams.

[26] Cirio, G., Marchal, M., Regia-Corte, T., and Lécuyer, A. The Magic Barrier Tape: a Novel Metaphor for Infinite Navigation in VirtualWorlds with a RestrictedWalkingWorkspace. *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology - VRST '09 1*, 212 (2009), 155.

[27] Cloudhead, G. VR Navigation, 2015.

[28] Collins, S., Ruina, A., Tedrake, R., and Wisse, M. Efficient bipedal robots based on passive-dynamic walkers. *Science (New York, N.Y.) 307*, 5712 (2005), 1082–1085.

[29] Collins, S. H., Adamczyk, P. G., and Kuo, A. D. Dynamic arm swinging in human walking. *Proceedings. Biological sciences / The Royal Society 276*, 1673 (2009), 3679–3688.

[30] Costanza, E., Inverso, S. A., and Allen, R. Toward subtle intimate interfaces for mobile devices using an EMG controller. *Proceedings of the SIGCHI conference on Human factors in computing systems CHI 05* (2005), 481.

[31] Crytek. The Climb.

[32] Darken, R. P., Cockayne, W. R., and Carmein, D. The omni-directional treadmill: A Locomotion Device for Virtual Worlds. *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST '97* (1997), 213–221.

[33] Depaula, R. A New Era in Human Computer Interaction : The Challenges of Technology as a Social Proxy. *CLIHC '03 Proceedings of the Latin American conference on Human-computer interaction* (2003), 219–222.

[34] di Pellegrino, G., and Ldavas, E. Peripersonal space in the brain. *Neuropsychologia 66* (2015), 126 – 133.

[35] DiZio, P., and Lackner, J. R. Circumventing side effects of immersive virtual environments. In *HCI (2)* (1997), 893–896.

[36] Drewes, H., De Luca, A., and Schmidt, A. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, ACM (2007), 364–371.

[37] Duchowski, A. T., Shivashankaraiah, V., Rawls, T., Gramopadhye, A. K., Melloy, B. J., and Kanki, B. Binocular eye tracking in virtual reality for

inspection training. In *Proceedings of the 2000 symposium on Eye tracking research & applications*, ACM (2000), 89–96.

[38] Duh, H. B.-L., Parker, D. E., and Furness, T. A. An independent visual background reduced simulator sickness in a driving simulator. *Presence: Teleoperators and Virtual Environments 13*, 5 (2004), 578–588.

[39] Essig, K., Pomplun, M., and Ritter, H. A neural network for 3D gaze recording with binocular eye trackers. *International Journal of Parallel, Emergent and Distributed Systems 21*, February 2015 (2006), 79–95.

[40] Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ACM (2015), 457–466.

[41] Esteves, A., Verweij, D., Suraiya, L., Islam, R., Lee, Y., and Oakley, I. SmoothMoves: Smooth Pursuits Head Movements for Augmented Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, ACM (New York, NY, USA, 2017), 167–178.

[42] Fernandes, A. S., and Feiner, S. K. Combating VR sickness through subtle dynamic field-of-view modification. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on*, IEEE (2016), 201–210.

[43] Fischer, S. L., Watts, P. B., Jensen, R. L., and Nelson, J. Energy expenditure, heart rate response, and metabolic equivalents (METs) of adults taking part in children's games. *Journal of sports medicine and physical fitness 44*, 4 (2004), 398.

[44] Fitzmaurice, G., Matejka, J., Mordatch, I., Khan, A., and Kurtenbach, G. Safe 3d navigation. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, ACM (2008), 7–15.

[45] Fove. The Eye Tracking VR Headset.

[46] Friedman, D., Leeb, R., Antley, A., Garau, M., Guger, C., Keinrath, C., Steed, A., Pfurtscheller, G., and Slater, M. Navigating virtual reality by thought: First steps. *Exploring the Sense of Presence* (2004), 160.

[47] Gamberini, L., Seraglia, B., and Priftis, K. Processing of peripersonal and extrapersonal space using tools: Evidence from visual line bisection in real and virtual environments. *Neuropsychologia 46*, 5 (2008), 1298 – 1304.

[48] Graziano, M. *The Spaces Between Us: A Story of Neuroscience, Evolution, and Human Nature*. Oxford University Press, 2017.

[49] Griffin, M. J. *Handbook of human vibration*. Academic press, 2012.

[50] Hall, E. T. The hidden dimension.

[51] Henze, N., Rukzio, E., and Boll, S. 100,000,000 taps: analysis and improvement of touch performance in the large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ACM (2011), 133–142.

[52] Hernandez Arieta, A., Katoh, R., Yokoi, H., and Wenwei, Y. Development of a multi-DOF electromyography prosthetic system using the adaptive joint mechanism. *Applied Bionics and Biomechanics 3*, 2 (2006), 101–111.

[53] Hernandez-Rebollar, J. L., Kyriakopoulos, N., and Lindeman, R. W. The AcceleGlove: a whole-hand input device for virtual reality. In *ACM SIGGRAPH 2002 conference abstracts and applications*, ACM (2002), 259.

[54] Hills, A. P., Mokhtar, N., and Byrne, N. M. Assessment of physical activity and energy expenditure: an overview of objective measures. *Frontiers in nutrition 1*, 5 (2014), 1–16.

[55] Holmes, N. P., and Spence, C. The body schema and multisensory representation (s) of peripersonal space. *Cognitive processing 5*, 2 (2004), 94–105.

[56] Ibayashi, H., Sugiura, Y., Sakamoto, D., Miyata, N., Tada, M., Okuma, T., Kurata, T., Mochimaru, M., and Igarashi, T. Dollhouse vr: a multi-view, multi-user collaborative design workspace with vr technology. In *SIGGRAPH Asia 2015 Emerging Technologies*, ACM (2015), 8.

[57] Interaction Design Foundation. The Glossary of Human Computer Interaction.

[58] Ishiguro, Y., Mujibiya, A., Miyaki, T., and Rekimoto, J. Aided eyes: eye activity sensing for daily life. In *Proceedings of the 1st Augmented Human International Conference*, ACM (2010), 25.

[59] Isokoski, P., and Martin, B. Performance of input devices in fps target acquisition. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*, ACE '07, ACM (New York, NY, USA, 2007), 240–241.

[60] Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1990), 11–18.

[61] Jacob, T., Bailly, G., Lecolinet, E., Casiez, G., and Teyssier, M. Desktop orbital camera motions using rotational head movements. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, ACM (2016), 139–148.

[62] Jacobsen, S. C., and Jerard, R. B. Computational requirements for control of the Utah arm. In *Proceedings of the 1974 annual conference-Volume 1*, ACM (1974), 149–155.

[63] Jaeger, B. K., and Mourant, R. R. Comparison of Simulator Sickness Using Static and Dynamic Walking Simulators. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 45* (2001), 1896–1900.

[64] Jaeger, B. K., and Mourant, R. R. Comparison of simulator sickness using static and dynamic walking simulators. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 45, SAGE Publications Sage CA: Los Angeles, CA (2001), 1896–1900.

[65] Jones, B. R., Benko, H., Ofek, E., and Wilson, A. D. IllumiRoom: peripheral projected illusions for interactive experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2013), 869–878.

[66] Kang, S. B. Hands-free navigation in VR environments by tracking the head. *Intl Journal of Human-Computer Studies 48*, 2 (1998), 247–266.

[67] Kasahara, S., Ando, M., Suganuma, K., and Rekimoto, J. Parallel eyes: Exploring human capability and behaviors with paralleled first person view sharing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM (2016), 1561–1572.

[68] Kasahara, S., and Rekimoto, J. JackIn: integrating first-person view with out-of-body vision generation for human-human augmentation. In *Proceedings of the 5th Augmented Human International Conference*, ACM (2014), 46.

[69] Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. Simulator Sickness Questionnaire: An Enhanced Method for Quantifying Simulator Sickness, 1993.

[70] Kennedy, R. S., Lane, N. E., Berbaum, K. S., and Lilienthal, M. G. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology 3*, 3 (1993), 203–220.

[71] Khan, A., Komalo, B., Stam, J., Fitzmaurice, G., and Kurtenbach, G. Hovercam: interactive 3d navigation for proximal object inspection. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, ACM (2005), 73–80.

[72] Kiguchi, K., Tanaka, T., and Fukuda, T. Neuro-fuzzy control of a robotic exoskeleton with EMG signals. *IEEE Transactions on fuzzy systems 12*, 4 (2004), 481–490.

[73] Kim, Y. R., and Kim, G. J. HoVR-Type: Smartphone as a typing interface in VR using hovering. In *Consumer Electronics (ICCE), 2017 IEEE International Conference on*, IEEE (2017), 200–203.

[74] Klimstra, M. D., Thomas, E., Stoloff, R. H., Ferris, D. P., Zehr, E. P., Klimstra, M. D., Thomas, E., Stoloff, R. H., and Ferris, D. P. Neuromechanical

considerations for incorporating rhythmic arm movement in the rehabilitation of walking Neuromechanical considerations for incorporating rhythmic arm movement in the rehabilitation of walking.

[75] Koller, D. R., Mine, M. R., and Hudson, S. E. Head-tracked orbital viewing: an interaction technique for immersive virtual environments. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, ACM (1996), 81–82.

[76] Konishi, Y., Hanamitsu, N., Outram, B., Minamizawa, K., Sato, A., and Mizuguchi, T. Synesthesia Suit. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 149.

[77] Lanier, J. Virtual reality: The promise of the future. *Interactive Learning International 8*, 4 (1992), 275–279.

[78] LaViola Jr, J. J. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin 32*, 1 (2000), 47–56.

[79] Lawson, B. D. Motion Sickness Symptomatology and Origins., 2014.

[80] Lee, J. S., Lee, S. E., Jang, S. Y., and Park, K. S. A simplified hand gesture interface for spherical manipulation in virtual environments. In *Proceedings of the 2005 international conference on Augmented tele-existence*, ACM (2005), 284.

[81] Lindenhof, B. Climbey.

[82] MacKenzie, I. S., and Soukoreff, R. W. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction 17*, 2-3 (2002), 147–198.

[83] Mackinlay, J. D., Card, S. K., and Robertson, G. G. Rapid controlled movement through a virtual 3d workspace. In *ACM SIGGRAPH Computer Graphics*, vol. 24, ACM (1990), 171–176.

[84] Maravita, A., and Iriki, A. Tools for the body (schema). *Trends in cognitive sciences 8*, 2 (2004), 79–86.

[85] McCullough, M., Xu, H., Michelson, J., Jackoski, M., Pease, W., Cobb, W., Kalescky, W., Ladd, J., and Williams, B. Myo arm: swinging to explore a VE. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception*, ACM (2015), 107–113.

[86] McIntosh, K., Mars, J., Krahe, J., McCann, J., Rivera, A., Marsico, J., Israr, A., Lawson, S., and Mahler, M. Magic Bench: A Multi-user & Multi-sensory AR/MR Platform. In *ACM SIGGRAPH 2017 VR Village*, SIGGRAPH '17, ACM (New York, NY, USA, 2017), 11:1–11:2.

[87] Mine, M. R. Isaac: a meta-cad system for virtual environments. *Computer-Aided Design 29*, 8 (1997), 547–553.

[88] Miniotas, D. Application of Fitts' law to eye gaze interaction. *CHI '00 extended abstracts on Human factors in computer systems - CHI '00* (2000), 339–340.

[89] Mohammad, Y., and Nishida, T. Using physiological signals to detect natural interactive behavior. *Applied Intelligence 33*, 1 (2010), 79–92.

[90] Moseley JR, J. B., Jobe, F. W., Pink, M., Perry, J., and Tibone, J. EMG analysis of the scapular muscles during a shoulder rehabilitation program. *The American Journal of Sports Medicine 20*, 2 (1992), 128–134.

[91] Myo. Myo Gesture Control Armband, 2013.

[92] Nash, D. The hyg database. *URL: https://github. com/astronexus/HYG-database* (2011).

[93] Nelson, N. Npr-all tech considered-virtual reality's next hurdle: Overcoming 'sim sickness', 2015.

[94] Norman, D. A. Natural user interfaces are not natural. *interactions 17*, 3 (2010), 6–10.

[95] Oculus. Introduction to Best Practices.

[96] Oculus. The Rift's Recommended Spec, PC SDK 0.6 Released, and Mobile VR Jam Voting, 2015.

[97] Ortega, M., Stuerzlinger, W., and Scheurich, D. Shocam: A 3d orbiting algorithm. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ACM (2015), 119–128.

[98] Outram, B. I., Pai, Y. S., Fan, K., Minamizawa, K., and Kunze, K. AnyOrbit: Fluid 6DOF Spatial Navigation of Virtual Environments using Orbital Motion. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, ACM (2016), 199.

[99] Pai, Y. S. Physiological Signal-Driven Virtual Reality in Social Spaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 25–28.

[100] Pai, Y. S., and Kunze, K. Armswing: Using arm swings for accessible and immersive navigation in ar/vr spaces. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, MUM '17, ACM (New York, NY, USA, 2017), 189–198.

[101] Pai, Y. S., Outram, B., Vontin, N., and Kunze, K. Transparent Reality: Using Eye Gaze Focus Depth as Interaction Modality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 171–172.

[102] Pai, Y. S., Outram, B. I., Tag, B., Isogai, M., Ochi, D., Kimata, H., and Kunze, K. Cleavr: Collaborative layout evaluation and assessment in virtual reality. In *ACM SIGGRAPH 2017 Posters*, SIGGRAPH '17, ACM (New York, NY, USA, 2017), 20:1–20:2.

[103] Pai, Y. S., Outram, B. I., Tag, B., Isogai, M., Ochi, D., and Kunze, K. Gazesphere: Navigating 360-degree-video environments in vr using head rotation and eye gaze. In *ACM SIGGRAPH 2017 Posters*, SIGGRAPH '17, ACM (New York, NY, USA, 2017), 23:1–23:2.

[104] Pai, Y. S., Tag, B., Outram, B., Vontin, N., Sugiura, K., and Kunze, K. GazeSim: simulating foveated rendering using depth in eye gaze for VR. In *ACM SIGGRAPH 2016 Posters*, ACM (2016), 75.

[105] Pai, Y. S., Yap, H. J., Dawal, S. Z. M., Ramesh, S., and Phoon, S. Y. Virtual Planning, Control, and Machining for a Modular-Based Automated Factory Operation in an Augmented Reality Environment. *Scientific reports 6* (2016).

[106] Pai, Y. S., Yap, H. J., and Singh, R. Implementation of a Voice-ControlSystem for Issuing Commands in a Virtual Manufacturing Simulation Process. In *Advanced Materials Research*, vol. 980, Trans Tech Publ (2014), 165–171.

[107] Pai, Y. S., Yap, H. J., and Singh, R. Augmented reality–based programming, planning and simulation of a robotic work cell. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 229*, 6 (2015), 1029–1045.

[108] Park, J. Synthesis of natural arm swing motion in human bipedal walking. *Journal of Biomechanics 41*, 7 (2008), 1417–1426.

[109] Patney, A., Kim, J., Salvi, M., Kaplanyan, A., Wyman, C., Benty, N., Lefohn, A., and Luebke, D. Perceptually-based foveated virtual reality. In *ACM SIGGRAPH 2016 Emerging Technologies*, ACM (2016), 17.

[110] Pausch, R., Burnette, T., Brockway, D., and Weiblen, M. E. Navigation and locomotion in virtual worlds via flight into hand-held miniatures. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques - SIGGRAPH '95* (1995), 399–400.

[111] Pausch, R., Shackelford, M. A., and Proffitt, D. A user study comparing head-mounted and stationary displays. In *Virtual Reality, 1993. Proceedings., IEEE 1993 Symposium on Research Frontiers in*, IEEE (1993), 41–45.

[112] Pavlik, R. A. *Enabling natural interaction for virtual reality.* PhD thesis, Iowa State University, 2014.

[113] Péruch, P., Belingard, L., and Thinus-blanc, C. Transfer of Spatial Knowledge from Virtual to Real Environments. *Spatial Cognition II*, 1998 (2000), 253–264.

[114] Pfeiffer, T., Latoschik, M. E., Wachsmuth, I., and Herder, J. Evaluation of binocular eye trackers and algorithms for 3d gaze interaction in virtual reality environments. *Journal of Virtual Reality and Broadcasting 5*, 16 (2008), 16.

[115] Phillips, C. B., Badler, N. I., and Granieri, J. Automatic viewing control for 3d direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, ACM (1992), 71–74.

[116] Pomplun, M., Ritter, H., and Velichkovsky, B. An Arti cial Neural Network for High Precision Eye Movement Tracking. 1–8.

[117] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. The Go-go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, ACM (New York, NY, USA, 1996), 79–80.

[118] Psotka, J. Immersive training systems: Virtual reality and education and training. *Instructional science 23*, 5-6 (1995), 405–431.

[119] Ramcharitar, A., and Teather, R. J. A Fitts' Law Evaluation of Video Game Controllers: Thumbstick, Touchpad and Gyrosensor. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, ACM (New York, NY, USA, 2017), 2860–2866.

[120] Razzaque, S., Swapp, D., Slater, M., Whitton, M. C., and Steed, A. Redirected Walking in Place.

[121] Rebenitsch, L., and Owen, C. Review on cybersickness in applications and visual displays. *Virtual Reality 20*, 2 (2016), 101–125.

[122] Rechy-Ramirez, E. J., and Hu, H. Bio-signal based control in assistive robots: a survey. *Digital Communications and Networks 1*, 2 (2015), 85–101.

[123] Ricciardi, E., Menicagli, D., Leo, A., Costantini, M., Pietrini, P., and Sinigaglia, C. Peripersonal space representation develops independently from visual experience. *Scientific reports 7*, 1 (2017), 17673.

[124] Riecke, B. E., Bodenheimer, B., McNamara, T. P., Williams, B., Peng, P., and Feuereissen, D. Do we need to walk for effective virtual reality navigation? Physical rotations alone may suffice. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6222 LNAI* (2010), 234–247.

[125] Rochester, N., and Seibel, R. Communication device, 1962.

[126] Ruddle, R. A., and Lessels, S. For efficient navigational search, humans require full physical movement, but not a rich visual scene. *Psychological Science 17*, 6 (2006), 460–465.

[127] Ruddle, R. A., and Lessels, S. The benefits of using a walking interface to navigate virtual environments. *ACM Transactions on Computer-Human Interaction 16*, 1 (2009), 1–18.

[128] Saponas, T. S., Tan, D. S., Morris, D., Balakrishnan, R., Turner, J., and Landay, J. A. Enabling always-available input with muscle-computer interfaces. *Proceedings of the 22nd annual ACM symposium on User interface software and technology* (2009), 167–176.

[129] Saraiji, M. Y. *Embodied-Driven Design: a Framework to Configure Body Representation Mapping.* PhD thesis, Keio University, 2017.

[130] Saraiji, Y. PupilHMDCalibration, 2016.

[131] Saraiji, Y., Sugimoto, S., Fernando, C. L., Minamizawa, K., and Tachi, S. Layered telepresence: simultaneous multi presence experience using eye gaze based perceptual awareness blending. In *ACM SIGGRAPH 2016 Posters*, ACM (2016), 20.

[132] Sasaki, T., Saraiji, M. Y., Fernando, C. L., Minamizawa, K., and Inami, M. Metalimbs: Multiple arms interaction metamorphism. In *ACM SIGGRAPH*

2017 Emerging Technologies, SIGGRAPH '17, ACM (New York, NY, USA, 2017), 16:1–16:2.

[133] Schmidt, A. Implicit human computer interaction through context. *Personal and Ubiquitous Computing 4*, 2 (2000), 191199.

[134] Schuemie, M. J., Van Der Straaten, P., Krijn, M., and Van Der Mast, C. A. P. G. Research on presence in virtual reality: A survey. *CyberPsychology & Behavior 4*, 2 (2001), 183–201.

[135] Seo, B.-K., Choi, J., Han, J.-H., Park, H., and Park, J.-I. One-handed Interaction with Augmented Virtual Objects on Mobile Devices. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '08, ACM (New York, NY, USA, 2008), 8:1–8:6.

[136] Serino, A., Noel, J.-P., Mange, R., Canzoneri, E., Pellencin, E., Ruiz, J. B., Bernasconi, F., Blanke, O., and Herbelin, B. Peripersonal space: An index of multisensory bodyenvironment interactions in real, virtual, and mixed realities. *Frontiers in ICT 4* (2018), 31.

[137] Shen, L., Saraji, M. Y., Kunze, K., and Minamizawa, K. Unconstrained neck: Omnidirectional observation from an extra robotic neck. In *Proceedings of the 9th Augmented Human International Conference*, AH '18, ACM (New York, NY, USA, 2018), 38:1–38:2.

[138] Slambekova, D., Bailey, R., and Geigel, J. Gaze and gesture based object manipulation in virtual worlds. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, ACM (2012), 203–204.

[139] Slater, M., Usoh, M., and Steed, A. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction 2*, 3 (1995), 201–219.

[140] SMI. SMI Gaze and Eye Tracking Systems, 2014.

[141] Smolic, A., Mueller, K., Merkle, P., Fehn, C., Kauff, P., Eisert, P., and Wiegand, T. 3d video and free viewpoint video-technologies, applications

and mpeg standards. In *Multimedia and Expo, 2006 IEEE International Conference on*, IEEE (2006), 2161–2164.

[142] Spano, L. D. A model-based approach for gesture interfaces. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '11, ACM (New York, NY, USA, 2011), 327–330.

[143] Stampe, D. M. Heuristic filtering and reliable calibration methods for video-based pupil-tracking systems. *Behavior Research Methods, Instruments, & Computers 25*, 2 (1993), 137–142.

[144] Stanney, K. M., and Hash, P. Locus of user-initiated control in virtual environments: Influences on cybersickness. *Presence: Teleoperators and Virtual Environments 7*, 5 (1998), 447–459.

[145] Sturman, D. J., Zeltzer, D., and Pieper, S. Hands-on interaction with virtual environments. In *Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology*, ACM (1989), 19–24.

[146] Suma, E. A., Clark, S., Krum, D., Finkelstein, S., Bolas, M., and Warte, Z. Leveraging change blindness for redirection in virtual environments. In *Virtual Reality Conference (VR), 2011 IEEE*, IEEE (2011), 159–166.

[147] Tan, D. S., Robertson, G. G., and Czerwinski, M. Exploring 3d navigation: combining speed-coupled flying with orbiting. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2001), 418–425.

[148] Terziman, L., Marchal, M., Emily, M., Multon, F., Arnaldi, B., and Lécuyer, A. Shake-your-head. *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology - VRST '10* (2010), 27.

[149] Tregillus, S., and Folmer, E. VR-STEP: Walking-in-Place using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (2016), 1250–1255.

[150] Tregillus, S., and Folmer, E. Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. In *Proceedings of*

the 2016 CHI Conference on Human Factors in Computing Systems, ACM (2016), 1250–1255.

[151] Treisman, M. Motion sickness: an evolutionary hypothesis. *Science 197*, 4302 (1977), 493–495.

[152] Ueda, Y., Asai, Y., Enomoto, R., Wang, K., Iwai, D., and Sato, K. Body Cyberization by Spatial Augmented Reality for Reaching Unreachable World. In *Proceedings of the 8th Augmented Human International Conference*, AH '17, ACM (New York, NY, USA, 2017), 19:1–19:9.

[153] Unity. Movement in VR, 2015.

[154] Unity. User Interfaces for VR, 2015.

[155] Uno, Y., Kawato, M., and Suzuki, R. Formation and control of optimal trajectory in human multijoint arm movement. *Biological cybernetics 61*, 2 (1989), 89–101.

[156] Usoh, M., Arthur, K., Whitton, M. C., Bastos, R., Steed, A., Slater, M., and Brooks, F. P. Walking {&}gt; walking-in-place {&}gt; flying, in virtual environments. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques - SIGGRAPH '99* (1999), 359–364.

[157] Valli, A. The design of natural interaction. *Multimedia Tools and Applications 38*, 3 (2008), 295–305.

[158] van der Meulen, H., Kun, A. L., and Shaer, O. What Are We Missing?: Adding Eye-Tracking to the HoloLens to Improve Gaze Estimation Accuracy. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, ACM (New York, NY, USA, 2017), 396–400.

[159] Venturino, M. Performance and head movements using a helmet-mounted display with different sized fields-of-view. *Optical Engineering 29*, 8 (1990), 870–877.

[160] Vrijkotte, T. G., van Doornen, L. J., and de Geus, E. J. Effects of work stress on ambulatory blood pressure, heart rate, and heart rate variability. *Hypertension (Dallas, Tex. : 1979) 35*, 4 (2000), 880–6.

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

[162] Westerterp, K. R. Assessment of physical activity: A critical appraisal. *European Journal of Applied Physiology 105*, 6 (2009), 823–828.

[163] Wexelblat, A. An approach to natural gesture in virtual environments. *ACM Trans. Comput.-Hum. Interact. 2*, 3 (Sept. 1995), 179–200.

[164] Williams, B., Bailey, S., Narasimham, G., Li, M., and Bodenheimer, B. Evaluation of walking in place on a Wii balance board to explore a virtual environment. *ACM Transactions on Applied Perception 8*, 3 (2011), 1–14.

[165] Wilson, A. D., and Cutrell, E. Flowmouse: A computer vision-based pointing and gesture input device. *HumanComputer Interaction-INTERACT 2005 3585* (2005), 565578.

[166] Wilson, P. T., Kalescky, W., MacLaughlin, A., and Williams, B. VR locomotion: walking¿ walking in place¿ arm swinging. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1*, ACM (2016), 243–249.

[167] Wilson, P. T., Nguyen, K., Harris, A., and Williams, B. Walking in place using the Microsoft Kinect to explore a large VE. *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry - VRCAI '14* (2014), 27–33.

[168] Witmer, B. G., and Singer, M. J. Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence: Teleoper. Virtual Environ. 7*, 3 (1998), 225–240.

[169] Yan, L., and Allison, R. S. New simple virtual walking method walking on the spot.

[170] Yan, L., Allison, R. S., and Rushton, S. New simple virtual walking method walking on the spot.

[171] Zander, T. O., Gaertner, M., Kothe, C., and Vilimek, R. Combining Eye Gaze Input With a BrainComputer Interface for Touchless HumanComputer Interaction. *International Journal of Human-Computer Interaction 27*, 1 (2010), 38–51.

[172] Zeleznik, R., and Forsberg, A. Unicam—2d gestural camera controls for 3d environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, ACM (1999), 169–173.

[173] Zhang, X., and MacKenzie, I. S. Evaluating Eye Tracking with ISO 9241 - Part 9. In *Proceedings of the 12th International Conference on Human-computer Interaction: Intelligent Multimodal Interaction Environments*, HCI'07, Springer-Verlag (Berlin, Heidelberg, 2007), 779–788.

[174] Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., and Harvill, Y. A hand gesture interface device. In *ACM SIGCHI Bulletin*, vol. 18, ACM (1987), 189–192.

List of Publications

1. Pai, Y.S., Chen, Z., Chan, L., and Kunze, K. PinchMove: Improved Accuracy of User Mobility for Near-Field Navigation in Virtual Environments. In *20th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM (2018).

2. Outram, B., Pai, Y.S., Person, T., Minamizawa, K., and Kunze, K. AnyOrbit: Orbital navigation in virtual environments with eye-tracking. In *ACM Symposium on Eye Tracking Research and Applications*, ACM (2018).

3. Pai, Y.S., Kunze, K. Armswing: using arm swings for accessible and immersive navigation in AR/VR spaces. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, ACM (2017), 189-198.

4. Pai, Y.S., Isogai, M., Ochi, D., Kimata, H., Kunze, K. face2faceVR: Using AR to Assist VR in Ubiquitous Environment Usage. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, ACM (2017), 173-176.

5. Pai, Y.S., Outram, B., Tag, B., Isogai, M., Ochi, D., Kunze, K. GazeSphere: Navigating 360-Degree-Video Environment in VR Using Head Rotation and Eye Gaze. In *ACM SIGGRAPH 2017 Posters*, ACM (2017), 23.

6. Pai, Y.S., Outram, B., Tag, B., Isogai, M., Ochi, D., Kimata, H., Kunze, K. CleaVR: Collaborative Layout Evaluation and Assessment in Virtual Reality. In *ACM SIGGRAPH 2017 Posters*, ACM (2017), 20.

7. Assilmia, F., Pai, Y.S., Okawa, K., Kunze, K. IN360: A 360-Degree-Video Platform to Change Students Preconceived Notions on their Career. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM (2017), 2359-2365.

8. Pai, Y.S. Physiological Signal-Driven Virtual Reality in Social Spaces. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 25-28.

9. Pai, Y.S., Yap, H.J., Dawal, S.Z.M., Ramesh, S., and Phoon, S.Y. Virtual Planning, Control, and Machining for a Modular-Based Automated Factory Operation in an Augmented Reality Environment. *Scientific Reports*, Nature Publishing Group (2016), 6.

10. Pai, Y.S., Tag, B., Outram, B., Vontin, N., Sugiura, K., and Kunze, K. GazeSim: simulating foveated rendering using depth in eye gaze for VR. In *ACM Siggraph 2016 Poster*, ACM(2016), 75.

11. Pai, Y.S., Outram, B., Vontin, N., and Kunze, K. Transparent Reality: Using Eye Gaze Focus Depth as Interaction Modality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 171-172.

12. Outram, B., Pai, Y.S., Fan, K., Minamizawa, K., and Kunze, K. AnyOrbit: Fluid 6DOF Spatial Navigation of Virtual Environments using Orbital Motion. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, ACM (2016), 199.

13. Phoon, S.Y., Yap, H.J., Taha, Z., and Pai, Y.S. Interactive solution approach for loop layout problem using virtual reality technology. *The International Journal of Advanced Manufacturing Technology*, Springer London (2016), 1-11.

14. Pai, Y.S., Yap, H.J., and Ramesh, S. Augmented realitybased programming, planning and simulation of a robotic work cell. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, SAGE Publications (2015), 1029-1045.

15. Yap, H.J., Pai, Y.S., Chang, S.W., and Yap, K.S. Development of an Augmented Reality-Based G-Code Generator in a Virtual CNC Milling Simulation. *International Journal of Computer Science and Engineering (IJCSE)*, 2015, 63-72.

16. Chew, S.S., Ghazilla, R.A.R., Yap, H.J., and Pai, Y.S. Development of Augmented Reality Approach Towards Ergonomic Assessment of Driver Vehicle

Package Design. In *3rd International Conference on Ergonomics & 1st International Conference of Industrial Engineering (ICE & ICIE 2015)*, 2015.

17. Chew, S.S., Ghazilla, R.A.R., Yap, H.J., and Pai, Y.S. Framework of Augmented Reality Approach Towards Ergonomic Assessment of Driver Vehicle Package Design. *Jurnal Teknologi*, 2015, 113-118.

18. Pai, Y.S., Yap, H.J., Ramesh, S., Chang, S.W., Cheong, K.L.R., and Taha, Z. Augmented Reality Assisted Factory Layout Planning and Analysis for a Flexible Manufacturing Cell. In *Proceedings of the 3rd International Conference on Computer Science and Computational Mathematics 2014 (ICCSCM 2014)*, Science and Knowledge Research Society (2014), 106-111.

19. Pai, Y.S., Yap, H.J., and Ramesh, S. Implementation of a Voice-ControlSystem for Issuing Commands in a Virtual Manufacturing Simulation Process. In *Advanced Materials Research*, Trans Tech Publications (2014), 165-171.