Master of Puppets

How to trick the human mind to explore an infinite world

Emil Magnar Kjenstad

Thesis submitted for the degree of
Master in Programming and Systems Architecture
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2023

Master of Puppets

How to trick the human mind to explore an infinite world

Emil Magnar Kjenstad

Master of Puppets

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Acknowledgements

First of all, I would like to give a special thank you to my advisor, Carsten Griwodz, for his help on this project. His guidance and numerous hours of discussion and feedback have guided me through what it takes to complete such a task. I would also like to thank Master Student Halvor Ringsby and Rahel Gaeb for their efforts on this research project.

I would also like to thank all the participants in the study, without whom I would have had no data to present and analyze in this thesis.

In addition, I would like to extend my gratitude to my friends, family, and loved ones for their continued support on this journey.

Finally, I would like to thank the University of Oslo for the opportunity to pursue my master's degree program.

Emil Magnar Kjenstad
University of Oslo

Abstract

The Greek philosopher Aristotle proposed thousands of years ago that the perception of reality resides upon what our senses tell us is real. The advancement of modern-day technology enables us to properly test whether his theory is correct. Virtual reality is a simulation of a three-dimensional environment with which we can interact as if it were the real environment. With this technology, users can disconnect themselves from the real environment and enjoy a computer-generated world that seems real. This comes with some flaws, however. For example, a user often employs a joystick or omni-directional treadmill to move around in the virtual environment. Many experience this as either an unnatural form of movement that disconnects us from the virtual world or to be expensive and may not be affordable for everyone. An inexpensive solution that feels natural is therefore desired. The most natural form of movement would be walking, but even this has its limitations since issues such as unknowingly walking into walls or other objects can become significant safety risks in the real environment. Moreover, movement in the virtual environment would be limited to what we have available in the real environment because movements are mapped between the virtual and physical environments as 1:1 — whatever you do in the real environment, you do in the virtual environment.

Redirected walking is a term used to describe the remapping of position and orientation between the real environment and the virtual environment. If this is done correctly, we can expand the available area in which we can explore a virtual environment. This is done by manipulating human senses to make a person perceive their movements in the virtual environment as ones they are performing in the real environment. We can achieve this by manipulating the camera in the virtual environment by injecting unnoticeable movements that redirect the user. We have emphasized three known methods of redirection in our research, namely, translation, rotation, and curvature. Our aim was to explore the extent to which redirected walking is achievable in various situations without it becoming noticed by users.

We have performed a study regarding the techniques mentioned above. Based on the responses of the participants in the study, our research successfully demonstrated that redirection is possible and, consequently, a practical approximation of user motion is possible as well. In addition, we were able to find statistical significance for both curvature gain and rotation gain, although not for translation gain. We also investigated the use of distractors and how they might affect the perceptual threshold and found statistical significance under this condition. Finally, our research examined how different virtual environments affect the noticeability of redirected walking. We used linear regression for this purpose and found that virtual environments in which users are more immersed can have higher detection thresholds.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Informal Problem Description

Large technological advancements have been made in recent history to create virtual realities that users can explore. Virtual reality headsets react to the motion of the head in real time, and mimic those moves accordingly in the virtual world in order to create the sense that the user is in that virtual environment. There are limits to this, however. In early releases, the user would use a joystick to move their characters around in the virtual environment. Despite changes in how the controls work (such as using a mouse and keyboard, or controller), one essential element remains the same to their old counterparts: for most systems, the user sits in a chair and moves their avatar. This feels unnatural and restrictive as it removes a key physical aspect away from the user as they are not physically walking.

One solution would be to use a 360 degree treadmill,[1] which allows the user to simulate walking, while staying in the same spot. These come in numerous varieties, including a ″frictionless″ one that requires the user to utilize a harness that does not generate a feeling of effort or resistance. There is also an ″omni-directional″ type with which the treadmill changes direction. This provides a decent feeling of resistance, but is too slow for natural motion. Moreover, this solution may be deemed expensive for some users.

Finding a solution to the movement issue that both feels natural and limits cost is therefore desired. The most natural form of movement available is to move in the virtual environment by the means of walking in the real environment. There are limitations to this as well, however, such as the physical space in the room where the user is located: the user might simply walk unknowingly into objects in the real world.

Redirected Walking is a term used to describe the ″perceptually unnoticeable virtual camera motion offset from a user′s movement″ [45] that can help resolve this issue. Nilsson et al. [25] study conducted at Purdue University illustrates different types of gains in redirected walking, highlighting rotation gains, translation gains, curvature gains, and bending gains.

---

[1] A treadmill that allows motion in any direction.

Rotation Gain Rotation gain involves having the user rotate shorter or greater distances to obtain their desired virtual rotation distance.

Translation Gain Translation gain refers to the distance covered when a user moves. This would mean, for example, that the user would cover 4 meters in the virtual environment by walking 2 meters in the real environment.

Curvature Gain Curvature gain involves forcing the user to walk in a curve in the real environment when they attempt to walk straight or in less pronounce curve in the virtual environment.

Bending Gain Bending gain refers to having the user make sharper turns in the real environment when they walk on a curved path in the virtual environment.

Our intention is to build a system to further investigate these types of gain and with which to explore how strongly we can manipulate users before (a) they notice it, (b) find it annoying, or (c) find it unacceptable. We may force these changes in movement in multiple ways. It can be done by means of, but not limited to, rotating the virtual environment in a matter unnoticeable to the user, sudden changes in the virtual scene (which we refer as attention-grabs), visual or tactile clues, or a mixture of the three. In addition, humans have a built-in ″dead man's switch″ when they lose their sense of sight or when there is nothing around them that can be used to orient themselves, so to speak — it has been demonstrated that humans walk in circles when their sensory system does not provide orientation, which is typically known from walking in the desert. We would like to explore how we can manipulate this proven fact and implement it in virtual reality.

1.2 Research Questions

Based on the above considerations, we will proceed by designing a system and conducting user studies concerning following questions:

- Do users notice the various redirected walking techniques?
- How strongly can we manipulate users before they notice?
- Do the different ways of using attention grabs decrease detection?
- Is the system applicable?

1.3 Summary of Contributions

The main contributions of this thesis may be summarized as follows:

- We have proposed and evaluated the perceptual threshold of translation gain, rotation gain and curvature gain under different conditions. In contrast to most research in this area, our evaluation emphasizes user acceptance above detection thresholds.

- We have obtained experimental results which indicate that users are not sensitive to discrepancies between their movements in the real environment and the virtual environment.

- Our experimental results also shows that users are less likely to notice differences in their movements when they are placed in a game-like virtual environment rather than a standard task-oriented virtual environment.

- Since the traditional perspective regarding redirected walking decouples the one-to-one mapping between the real environment and the virtual environment, we can alter movements from the one environment to the other. The redirected walking algorithm involves a manipulation of the sensory system that makes the user perceive virtual reality as real. The user thus observes their virtual movements as their actual movements.

- The dominant view of redirected walking is concerned with the highest possible threshold of redirection. We claim that this will not work in a practical setting since our results indicate that users are sensitive to applied redirection in a manner that depends on the situation. When users are able to move freely over long distances, such as in an open world environment, they tend to be less sensitive to redirection. However, users can accurately adapt their movements when precise movements are required.

- Our model concerns users who have a finite space to work with in the real environment. In addition, it is not realistic to expect video game companies, or others who develop virtual reality applications, to implement redirected walking that would be suitable for their applications because of cost limitations. It is thus desirable to develop a generic algorithm that works for most situations.

1.4 Overview of Thesis

This thesis is divided into 7 chapters, including this introduction.

Chapter 2 Redirected walking employs various methods to manipulate human senses in order to achieve redirection. Consequently, it is important to understand the different senses humans have and how we can manipulate them. In this chapter we discuss the various human senses that can be used to achieve redirection and why this is the case.

Chapter 3 The first articles mentioning redirected walking were published in the early 2000s. Since then, numerous researchers have further investigated the various ways in which we can redirect the user. This chapter addresses this research and asks how we can use this work for our thesis.

Chapter 4 Before any lines of codes can be written, a theoretical approach to how we would like to conduct our research should be adopted. Doing so will limit problematic issues that may arise during development, set a common goal, and provide a better understanding of what we would like to achieve with our research. This chapter presents the design phase of our project, including how we would like to conduct our research.

Chapter 5 The problem of redirected walking can be solved in a multiple of ways. This could involve seamless rotations in the virtual environment or warping of the virtual scene. The choice of hardware is also essential to providing an optimal virtual reality experience. This chapter addresses the experimental setup of our research.

Chapter 6 No researcher can present their findings before conducting an analysis of the data collected. This chapter examines the data we collected during our research and interprets them by means of statistical analysis.

Chapter 7 This chapter concludes the thesis with a set of final remarks and provides suggestions for future research.

Chapter 2

The Sensory Nervous System

Redirected walking is concerned with making the user believe that they have walked a certain path when they have in fact walked a different path. This initially appears to be a difficult task to carry out successfully. However, studies show that humans naturally walk in a circular path under certain circumstances [40], which acts similar to a ″dead man's switch.″ This is reflected in J.R.R. Tolkien's *The Lord of the Rings: The Two Towers* (1954), for example, when the main character becomes lost and discovers he has been walking in circles. This not only is for dramatic effect, it is also based on the fact that humans do indeed walk in circles when there are no landmarks to guide them [32, 40]. There are similar trends in walking when we are visually impaired. For instance, people have tried walking in a straight line while blindfolded and discovered that they have walked in an arc. This is quite subjective, however, since certain people can walk in a straighter line while others walk in a significant curve.

One benefit of this is that it can be exploited by virtual reality. The reason is that wearing virtual reality goggles removes the ability to process the outside world, which leads to the ″dead man's switch″ mentioned above kicking in naturally. Our challenge then comes down to being able to create a tighter arc than what is natural for the user. That is not to say that an arc is the only pattern we can manipulate. As will be discussed in Chapter 3, we can manipulate the straight-line velocity and circular rotation as well. Furthermore, virtual reality is all about removing yourself from reality and placing yourself into a virtual world. To fully enjoy the virtual world that you are experiencing, you must believe that you are there. Creating this immersion makes the virtual world more believable and disconnects you from reality.

Our aim is to preserve immersion into the virtual world during reorientation. In order to do so, we must investigate what makes something that is believed into something that is real. Philosophers have long discussed what makes something that we perceive to be taken as real. For the purpose of this thesis, we will follow what the Greek philosopher Aristotle defined as real. Aristotle maintained that our senses are the building blocks for determining whether something is real, and that how we process the pieces of information that we obtain from our senses is what determines whether or not we take something to be is real [6]. If Aristotle's claim is correct, it would mean that we can make someone believe that something is real by manipulating their senses.

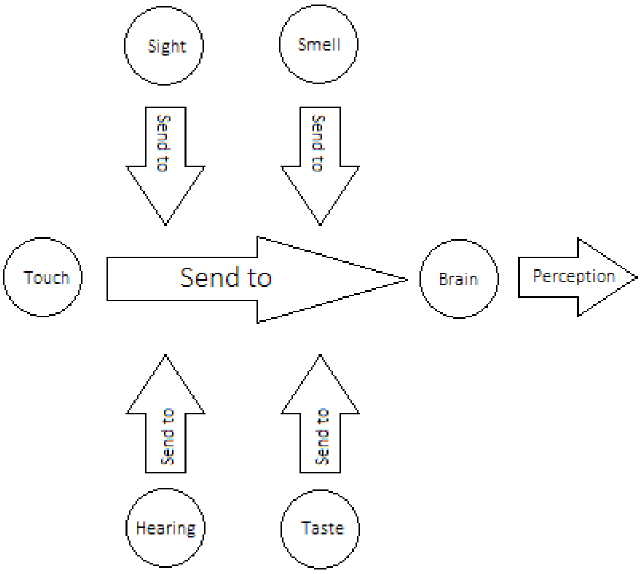


Figure 2.1: Flow of sensory information

The various senses that all humans have form our sensory nervous system. Each sense is tasked with registering different inputs from the environment within which we find ourselves. These bits of information are then pipelined to the human brain, where they are then processed and interpreted. The way in which we interpret our senses creates the perception that we have of the world around us. This process is illustrated in Figure 2.1.

Medical Term	Everyday Term
Visual system	Sight
Auditory system	Hearing
Somatosensory system	Touch
Gustatory system	Taste
Olfactory system	Smell

Table 2.1: Aristotle's five senses

Aristotle describes five senses that contribute to the human sensory system, namely, sight, hearing, touch, taste, and smell (see Table 2.1). These are the five senses we take into consideration in this thesis. Additional sensory systems have been defined since Aristotle put forward his claim, two of the most notable being the vestibular system (balance) and the interoceptive system (sensations from inside the body) [28]. The vestibular system could be particularly useful for our research in that it provides information about the body's position. As noted above, redirected walking is all about making the user believe that they have walked a certain path when they have in fact walked a different path. Consequently, if we could find a way to manipulate the vestibular system, then we would have found our solution. Simulating balance might prove to be difficult, however.

We brainstormed a number of ideas in this regard, but all of them involved additional hardware. This is not ideal, however, since the contraptions involved would be expensive, which is something that we are trying to avoid. For example, a floor mat that raises and lowers itself to slightly nudge the user is one possibility, but it would be so costly that it would not be useful to pursue it. Since solutions for tricking the vestibular and interoceptive systems are also too expensive and/or not feasible, we have not given consideration to sensory systems other than Aristotle's five senses. This chapter is therefore broken down into six sections, one for each sensory system that Aristotle describes and a final summary.

- Visual System (Section 2.1)
- Auditory System (Section 2.2)
- Somatosensory System (Section 2.3)
- Olfactory System (Section 2.4)
- Gustatory System (Section 2.5)
- Chapter Summary (Section 2.6)

The aim of this chapter is to gain a better understanding of the human sensory nervous system. We will discuss each of the five sensory systems — what it is, how we may manipulate it to achieve immersion in virtual reality, whether the means of manipulation are theoretically feasible, and how we can utilize manipulation within the context of redirected walking. In the final section, we will determine how we may implement such manipulation within the scope of this thesis.

2.1 Visual System

The visual system consists of the eyes and the various parts of the nervous system that give us sight and create a visual perception of the world. This includes receiving, processing, and interpreting visual information to build a representation of the visual environment. Our visual system is also tasked with gathering information about how our position changes in respect to the objects around us [13].

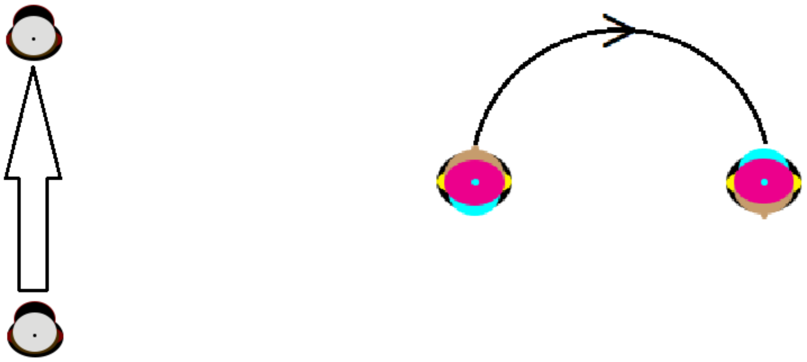
Manipulating the visual system is quite straightforward — since the visual system is based upon what we see, all we need to do is change what we are seeing. Moreover, technological advancements have provided us

Figure 2.2: The Visual System

with many devices that can do this for us. For example, consumers today can purchase virtual reality goggles that display a virtual environment to the user. All the user must then do is wear such goggles in order for us to succeed in manipulating their visual sense. We can display whatever we would like to the user with virtual reality goggles — a scene from your favorite movie, being in space, a different country, or an imaginary world. These are just a few examples of the different types of environments we can display. Our only limitation is our imagination.

The idea of using virtual reality goggles might at first seem discouraging if we are concerned with price. The issue of cost was in fact the reason we dismissed any attempt to manipulate the Somatosensory System (see Section 2.3), Olfactory System (see Section 2.4), and the Gustatory System (see Section 2.5). However, consumers in today's market can purchase virtual reality goggles somewhat inexpensively. In addition, the basic hardware needed for virtual reality would be the use of virtual reality goggles. This fact alone would make manipulating the visual system feasible insofar as it is in some regard the bare minimum needed to explore virtual reality.

Redirected walking is about changing the user's orientation in the real environment so they can explore an infinitely large virtual environment. One basic example of how visual manipulation would work in respect to redirected walking would involve having the virtual reality character walk in a straight line forever. To do so, the user would walk in a circular pattern in the real environment, while the virtual reality character would seem to walk in a straight line. Figure 2.3 compares a walking motion between the virtual environment and real environment, with both figures facing in the same direction in the starting position. The head is facing in the same direction in the final position in Figure 2.3a, but in the opposite direction in the final position in Figure 2.3b. This would mean that real world scene has rotated 180 degrees while the virtual scene has not rotated at all. If we want the user to believe that they are walking in a straight line when they are in fact walking in a circle, we need to rotate the virtual scene accordingly.



(a) User's movement in virtual environment

(b) User's movement in the real environment

Figure 2.3: Comparing movement in real time with virtual reality

We can also force the user to subconsciously walk in a circular path by rotating the virtual environment. If this is done correctly, the user unknowingly observes these tiny rotations as mistakes and subconsciously corrects their trajectory. The outcome is that the user can intuitively follow a virtual path much longer than the real environment permits — they walk in a straight line in the virtual environment while following a circular trajectory in the real environment [5, 13, 32, 42].

2.2 Auditory System

The auditory system enables us to perceive and interpret sounds. An object creates sound waves that are picked up by our ears, where the information of those sound waves are registered and processed. With our auditory system, we can tell what direction a sound came from in a three-dimensional space and what the sound was. In addition, the use of audio can be used to instigate different reactions. This is heavily utilized in film, where sounds such as music and various sound effects help establish the mood of the scene and a sense of reality [10] — heavy rain is often sad, a fun song implies a good time, etc. A less frequently used aspect of audio is that it can discern distances involving objects around us. Echolocation is one means for a blind person to navigate around obstacles using reflected sounds.

Making someone hear something that is not in the real environment is quite easy — all we would need to do is create sound waves that the user can pick up. The most basic example of this, but also the most realistic, would be the use of headphones or speakers. These devices create sound waves based on some input — a song from your music library, noises from your video game, sounds from a movie, etc. By means of headphones and an input device, we can thus simulate sounds in the virtual environment and manipulate the auditory system.

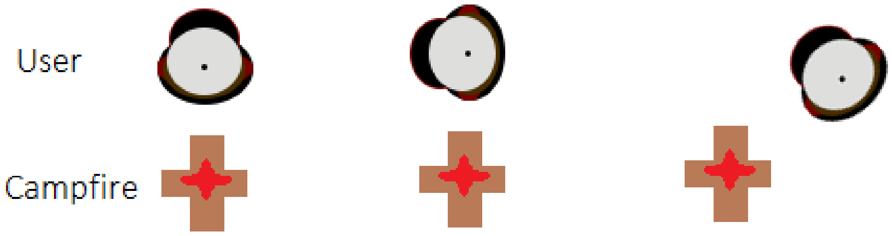
Headphones in today's market range from 5 USD for the most basic

Figure 2.4: The Auditory System

models to expensive ones that cost 300 USD, any of which produce sounds that we can hear. Most virtual reality goggles also come with built-in speakers or headsets that create sounds for the user, which means that the equipment issue is easily resolved in this regard. In respect to the actual sounds we want to produce, numerous game engines allow us to create sources of sound and place them in the virtual world. When the user then enters the virtual world, the headset will use these sound sources and provide the user with a full 360 degree surround sound experience of the environment. With this in mind, it should be quite feasible to manipulate the auditory system.

Audio can be used to give a sense of where something or someone is located in a room. Children often play a game in the dark called hide and seek, also known as Marco Polo, in which one of the participants is the seeker whom the others try to avoid. The seeker shouts ”Marco,” to which the others are to reply ”Polo,” and then uses this audio cue to determine where the others are and try to capture them. The seeker may also hear someone walking close by or breathing heavily near them, which can give a sense of where they. If we use this game to illustrate a theoretical concept, we see that it should be possible to redirect the user using audio cues. Let us assume, for example, that the user is visually impaired and is standing near a campfire. The campfire is a source of sound to the user, and they can make assumptions about where they are on the basis of that source. As the user moves around, it should thus be possible to move the sound of the campfire in order to reorient the user. For instance, the user may want to turn 90 degrees to the left in the virtual world. If the sound source is initially located directly behind them (Figure 2.5a), they can expect the sound to be on their left when they have finished turning (Figure 2.5b). We should therefore be able to turn the user more or less than 90 degrees, depending on what we would like to achieve, by over or under adjusting the placement of the sound source (Figure 2.5c).

Audio is thus a second key element in fully immersing the user in the



(a) User's initial position in virtual and real world

(b) User's final position in virtual world

(c) User's final position in the real world

Figure 2.5: Comparing rotation in real time to virtual reality using audio

virtual environment. However, conflicting audio and visual stimulations between the virtual and real environments may reveal to the user where they are positioned and break their immersion. The blindfold maze is a game that helps illustrate this point. This game pairs two people together, one who is blindfolded and an observer. The blindfolded person is placed inside a maze and they try to find a way out. The observer, who is looking at the maze, gives directions to the person who is blindfolded. The one who is blindfolded knows where they are in relation to the observer while the observer is calling out directions. When we translate this into redirected walking, we see that the user may know where they are in the real environment if some outside noise makes it through to them. A sound source they constantly hear, such as the fans from a computer, could ruin the immersion. Ruining the immersion could in turn ruin redirected walking since the user may know on the basis of outside noises that they did not make the move in the real environment that was translated into the virtual environment.

2.3 Somatosensory System

The somatosensory system is commonly known as touch perception, which enables us to distinguish between a wide range of different objects. Every day our bodies are exposed to numerous types of contact and textures, and yet we have no difficulty in distinguishing the different forms of contact from each other and react appropriately to each of them. A gust of wind, heat from the sun, being poked on the shoulder, and scratching itchy skin are just a few of the forms of contact that we are exposed to in daily life. In general, there are two main contributors to differentiating between the variety of tactile touches: (1) glabrous or hairless skin and (2) hairy skin [1]. Glabrous skin is commonly associated with finger tips, the palms of our hands, and the bottoms of our feet. These areas are more sensitive than other in registering an object's physical attributes, such as texture and shape. For example, we normally use our finger tips and hands for direct contact with other objects more than any other parts of our bodies, and it would therefore make sense that they are more sensitive to physical contact. Hairy skin, on the other hand, has less direct contact with an object. The areas of our bodies with such skin have some of the same abilities as glabrous skin, but the number of cells that register direct contact are significantly fewer. To compensate, we have hair that is sensitive to other forms of contact, and various cells associated with such hair are sensitive to the hair's movement. Examples of the information they can register include movements of the air, scratching, and poking [1].

Figure 2.6: The Somatosensory System

We would have to simulate two types of stimuli if we wished to manipulate our touch perceptions. These would be (1) direct contact and (2) indirect contact. Indirect contact occurs when the object causing a stimulus is not in contact with the person, such as when air blows over a person's skin. One way to simulate this would be to simply have an array of fans that blow air at the user from different directions and with different speeds to simulate wind. Direct contact occurs when an object is in physical contact with the person, including poking, scratching, holding an object, rain, etc. Mimicking these types of scenarios would consequently involve being in physical contact with the person. This type of technology is not something new. For example, instead of going to a massage therapist, today we can purchase massage chairs and pillows that create similar sensations. There have also been large advances in haptic feedback, which refers to communicating through the use of touch. A vibrating phone would be an example of haptic feedback in that the vibrations communicate to the user that they have a new notification. On a larger scale, haptic feedback could also be something that adds resistance to a specific movement to simulate weightlifting. Technological advancements have produced haptic feedback suits that could be a good solution in simulating direct contact.

Haptic feedback suits and an array of fans could thus be a valid solution. Fans are already used in film studios to simulate blowing wind. If we obtain such fans and arrange them accordingly, then we could simulate air blowing from whatever direction we desire. Haptic feedback suits that we can use to simulate all types of direct contact are also on the market and available to the consumer. It thus appears quite feasible to simulate touch perception.

However, although fans and haptic feedback suits might be useful for immersion into virtual reality, we see no obvious way how they would benefit redirected walking. Matsumoto et al. [20] demonstrate that redirection can be performed through the use of touch, but they refer to constant touch in the real environment, not in the virtual environment. Utilizing fans and a haptic feedback suit thus loses its value for redirected walking. It is also the case that some forms of touch perception are undesirable, such as the numerous electrical and signal cables that must be managed in virtual reality. They could give away the user's orientation and position in the real environment if they were accidentally stepped on, came into contact with one's body, or pulled a person in one way or another because of their weight. Experiencing these sensations from the real environment could break the immersion and ruin the attempt at redirected walking [42].

2.4 Olfactory System

The olfactory system is the sense of smell, which enables us to a certain degree to establish what the smell is and where it comes from. We are able to distinguish numerous smells from each other — strawberry, vanilla, bacon, and many, many more. Simply put, we detect different types of smell when inhaling air that contains odorant molecules. These molecules transfer information by making contact with receptors associated with the olfactory system. The information received is then sent to the brain to be processed [35]. However, one such molecule does not contain all the information we use to make out what a smell is. A given smell is instead made up of hundreds of different odor molecules coming together [48].

Research on the olfactory system appears to lag behind research on the other senses [35]. There is consequently much speculation and debate about how we can manipulate the sense of smell. We noted in Section 2.3 that we could use an array of fans as a point source for creating airflow in order to simulate indirect contact. Since we can also detect the direction smells come from, a similar solution could be explored in respect the olfactory system. If we had some type of devices that artificially create different odors, we could create an array of such devices and place them in a circle around the user. They could then serve as point sources that would (a) create different smells the user could pick up and (b) guide the user to some degree.

Manipulating the olfactory system might work in theoretical terms. The question is whether there is a machine that can artificially create numerous types of smells. This seems to be the case in that consumers can purchase scratch and sniff technology, which refers to scratchable stickers that release different scents. The stickers themselves are coated with a fragrance that is released when scratched. We thus know that there is a type of machine

Figure 2.7: The Olfactory System

that can mimic different smells. Using such technology and significantly increasing the strength of the smell would mean that we could theoretically mimic smells in the virtual world.

One thing redirected walking does is steer the user away from danger, such as striking an object in the real environment. The method of redirecting the user should thus yield instant results since otherwise the time to redirect may be too long to prevent the user from walking into a problem. A potential flaw with using the olfactory system for redirected walking would be that. The time it takes for a scent to be registered is significantly longer than it is for visual or audio stimuli. Gas containing odorant molecules travels through the air at much lower speeds than a sound wave [47] — it takes a period of time before we start to smell the incense that we just began to burn. We therefore need to discuss how smell could assist us in redirected walking on the assumption that smell travels similarly to sound. We might proceed by examining a solution similar to the one we discussed in Section 2.2, the only difference being that we would replace different sounds with different scents. However, since there is little evidence indicating that people navigate on the basis of scents, we find it unrealistic to expect such a solution to work. On the other hand, people react to various smells when they stand out. The smell of food when someone is hungry or that of old garbage when someone lifts a garbage can lid are things that people may react to in a split second by looking around to see where the smell comes from. This split second provides an opportunity for us to reorient the user. We will discuss this in greater depth in Section 3.2.

2.5 Gustatory System

The gustatory system, which refers to the sense of taste, is comprised of taste cells that can indicate different types of taste. Five distinct types of taste have been identified, namely, salty, sweet, bitter, sour, and umami [40], and it is the numerous taste buds on our tongue that distinguish between them. It is known, however, that tastes do not rely solely on gustatory stimuli — our perception of how something tastes in fact relies upon how different senses interact with each other [29, 41]. For example, Murphy and Cain [23] state that the sense of smell enhances the flavor of what we are eating, and that if we were to block our sense of smell, it would reduce flavor to a significant degree. In addition, we notice the texture of our food and any potential sounds that it makes while eating and register how crunchiness and juiciness, for example, affect the flavor of our food [8, 14, 38]. Finally, how something looks also affects how something tastes. Laan et al. [16] write that sight is a visual cue for the body to prepare for food, and that it invokes physiological, emotional and cognitive responses that affects how something tastes.

Figure 2.8: The Gustatory System

There are numerous ways in which we could manipulate the gustatory system, some of which have already been mentioned in the paragraph above. These involve manipulating the gustatory system by sending false information to our other senses. Nakano et al. [24] mention other examples of how we could manipulate this sense:

- Stevenson, Prescott, and Boakes [44] discovered in their research that the taste of sucrose increased and that of citric acid decreased when respondents were presented with the odor of sweet caramel.
- Slocombe, Carmichael, and Simner [37] report that participants in their research observed stronger acidity in rough foods than in smooth foods.
- Little is needed to manipulate our sense of taste through vision. Simply altering the color of a food item resulted in participants reporting other flavors than what was given to them [30]
- A food simulator, which can reproduce both food texture and how hard you need to chew, has been developed by Iwata et al. [8].

It is thus possible to manipulate the gustatory system indirectly. While the research conducted in preparing this thesis uncovered numerous ways in which we could manipulate the sense of taste by feeding misinformation to the other senses, little was discovered concerning how we could give false information to our taste buds and manipulate our sense of taste directly. It in fact remains difficult at this moment to clarify whether we could manipulate the gustatory system directly. It was necessary in all cases, however, to feed the user some sort of consumable. This might prove to be an obstacle for virtual reality since the user must be aware of where their ″fake food″ is in the real environment if they are to eat it at the appropriate moment to simulate eating in the virtual environment. If there was a good way to create ″fake food″ for consumption, and if there was a good way for carrying this food around so we have it accessible, sense manipulation in this might be theoretically possible.

Employing this type of manipulation could be useful for fully immersing the user in the virtual environment — the user could eat items from the virtual environment and taste them. However, we do not see how this would assist our research. Our findings do not indicate that consuming something edible can change how the user observes their orientation in the real environment in comparison to the virtual environment.

2.6 Chapter Summary

In this section we will take into consideration the previous discussion and determine which sensory systems we will proceed with in this thesis.

Visual System We suggest that manipulating the visual system plays a significant role in respect to redirected walking. For the reasons discussed in Section 2.1, we will take into account the visual system when implementing our redirected walking algorithm. In addition, since the equipment involved is a basic necessity for virtual reality, the cost involved is not a significant factor.

Auditory System It should be theoretically possible to navigate a user in virtual reality solely by means of audio cues. Since most virtual reality headsets come with speakers that provides 3D spatial sound, paying for the necessary equipment is not an issue. We believe that in instances where manipulating the visual system is less effective, such as being in a dark room, redirected walking by means of audio alone should be explored. However, since we unfortunately do not have the time needed to explore such mechanics in this thesis, we will not include this as a method of redirection. We saw, however, the effectiveness of audio to obtain user′s attention in games such as Marco Polo. We are thus interested in exploring ways we can use audio as a form of attention grab.

Somatosensory System Although we have seen that it is feasible to redirect using the somatosensory system, we believe that it is not practical. Furthermore, acquiring the haptic feedback suit and fans needed for full immersion is expensive. Since our goal is to create an inexpensive solution that anyone could implement, the added cost of such products is beyond what we are considering. However, insofar as Steinicke et al. [42] show the importance of cable management, we propose that some type of wireless solution is desirable. This would permit the user to maneuver around freely without having cables in their way. A backpack for carrying the needed equipment or wireless virtual reality goggles would perhaps be sufficient. We will address this question below in Section 5.1.2.

Olfactory System It is hard to justify focusing on the olfactory system in this thesis. We find smell to work best as a method for simulating a user's specific environment (a farm, for example) or grabbing their attention (the smell of something good cooking). However, the cost of the equipment necessary is once again a significant problem. In addition, we based our redirected walking algorithm on the assumption that smell could travel as fast as sound, but this is not the case — smell travels too slow to be a viable solution. Consequently, we will not look further into manipulating the olfactory system for this thesis.

Gustatory System We do not see how exploring the gustatory system could provide anything of benefit for our thesis research. Manipulating this system would involve manipulating all of the other senses that we have examined in this chapter, and we have already dismissed attempting to implement some of them for various reasons, primarily the expense involved. These issues have been discussed in Section 2.3, and Section 2.4, and it would make no sense to address them again here. This is one reason for giving no further consideration to manipulating taste through smell and touch. Perhaps more importantly, the gustatory system does not help with redirected walking. Due to limited time and resources, concerning ourselves with a system that is not useful for the overall research would serve no purpose. For that reason alone, we will dismiss any further attempts to manipulate taste.

Chapter 3

Related Work

In this chapter we will look more closely at what redirected walking is. The term *redirected walking* was first described in 2001 by Razzaque, Kohn, and Whitton [32], who are viewed as some of the creators behind the first redirected walking algorithm [13]. Although walking is the means most often addressed for achieving full immersion in virtual reality, it restricts virtual space to a given physical space. Redirected walking translates walking in a physical space into movement in virtual space, and it expands the size of the virtual space past what is physically available [32]. In order to achieve this, we manipulate the camera movement in the virtual environment in respect to the tracked movements in the real environment. Virtual reality would traditionally map the user′s movement from the real world to the virtual world in a 1:1 ratio, which would mean that moving 1 meter in the real environment would move the user 1 meter in the virtual environment in the same direction. Manipulation of the camera movement decouples the 1:1 mapping between the two environments in respect to the user′s position and orientation. This yields a redirection from the intended virtual path that results in the actual walked path. We can thus explore a larger area through walking in the virtual environment than we can by walking in the limited space of the physical playing area would permit if we applied 1:1 translation [13]. The term redirected walking thus describes a ″perceptually unnoticeable virtual camera motion offset from a user′s movement″ [45].

The original redirected walking algorithm developed by Razzaque, Kohn, and Whitton [32] is based on seamlessly rotating the virtual environment in one direction. The user then unknowingly rotates their heads and torso with the rotation to ″correct″ themselves, causing redirection [32]. Newer methods to achieve redirected walking have been investigated since redirected walking was created. Continuous visuo-haptic interaction and velocity-dependent dynamic redirected walking, as discussed in Matsumoto et al. [20] and Kloos et al. [13] respectively, are two examples of such methods. We examined different versions of redirection in more detail later in this chapter.

For reasons discussed in Chapter 2, we mainly focus on redirected walking techniques that manipulate the visual system. This thesis primarily present the research we have conducted for this purpose. By manipulating the visual system, we include any mechanism that distorts and/or rotates the virtual scene.

Redirected walking is a research topic within the field of computer science, and the earliest article on this topic is that of Razzaque, Kohn, and Whitton [32] noted above. As with any other new research topic, it begins by researching its basics. Stated otherwise, the majority of the research conducted while writing this thesis targets only one specific element within redirected walking. However, we can quite easily see the different mechanisms that are now being used in redirected walking and can place them in one of two subcategories. The first contains methods of how we can redirect the user, while the second includes the various forms of how we can capture the user's attention in the virtual environment. Both subcategories are discussed in depth in this chapter, which consists of two sections, each of which is devoted to one of these two subcategories:

- Reorientation Techniques (Section 3.1)
- Attention Grabs (Section 3.2)

Since there is no use to reinvent the wheel by researching the same topics that have already been discussed at length, we will spend some time looking at what has been done so we can build upon it. The goal of this chapter is therefore to gain a better understanding of what redirected walking is and the existing research associated with it. We saw in Chapter 2 that redirected walking is fundamentally about manipulating our senses. We further concluded that we will mainly focus on the visual system for this thesis. We can therefore amend our original statement to include the discussion of the different ways in which we can manipulate the visual system to achieve redirection. Section 3.1 addresses the different forms of redirection we have for breaking the 1:1 mapping between the virtual and real environment. Section 3.2 addresses the different methods used to capture users' attention in order to steer them from danger and/or redirect them.

3.1 Reorientation Techniques

Reorientation techniques are the various techniques we can use to redirect the user in redirected walking. Any method that changes the user's path in virtual reality in respect to their walked path in the real world may therefore be regarded as a redirection technique. While there are numerous ways in which to redirect the user, all of them are based on one of two methods [25, 45]. The first changes how we map user position and rotation between the two environments by scaling the user's movements [25]. The second, which may be referred to as warping, changes the layout of the virtual environment in order to compress larger virtual environments into smaller finite areas [25, 46]. To our knowledge, warping the virtual scene is dependent upon prior knowledge of the virtual environment.

Information, such as locations, doors, and hallways, is needed to warp the virtual environment to fit the tracked walking area in the real environment. Making a generic solution that uses warping would therefore be quite difficult. On the other hand, the most basic scaling-based implementation, namely, steer-to-center, only depends on where the center of the real environment is. Since any virtual environment would work with this implementation, it appears to be more generic. For this reason, we chose to focus on scaling-based implementation.

Nilsson et al. [25] summarize the first 15 years of redirected walking in their research, highlighting four different types of ″gains.″ To our knowledge, there are no other types of gains than the four that they mention. Each technique serves its own purpose and can act independently of the others. It should be possible in theory, however, to use them in conjunction with one another in order to obtain an optimized redirected walking algorithm.

We have therefore divided this section into four subsections, one for each technique Nilsson et al. [25] have described:

- Translation Gain (Section 3.1.1)
- Rotation Gain (Section 3.1.2)
- Curvature Gain (Section 3.1.3)
- Bending Gain (Section 3.1.4)

We will discuss the basic idea of each of these four techniques, the formula used to calculate the scale on which we can reorient the user, the detection threshold,[1] and how the use of this technique changes the mapping between real environment and virtual environment.

3.1.1 Translation Gain

The velocity at which the user moves is one form of movement that we can manipulate. In the terminology of physics, velocity is the speed at which something travels in a given direction. The redirected walking algorithm takes the velocity of the user′s movement as an input and proceeds to scale it by a certain factor. We should note that certain researchers have suggested that we should perform scaling only in the user′s direction of walking in order to prevent scaling any unintended lateral movements [7, 15, 27]. Nonetheless, scaling makes it possible for us to take the user′s movements in the real environment and make them faster or slower in the virtual environment. The factor could either be a fixed variable, or dynamically change throughout the lifespan of the algorithm. In both cases, the resulting velocity vector is used to move the user in the virtual environment. This enables us to map a larger virtual environment to a smaller real environment [5, 25].

If the virtual environment is realistic, we have an approximate, intuitive understanding of size. Talking about ′1 meter′ in the virtual environment

---

[1] Manipulation strength before becoming noticeable to the user.

means an approximation with respect to intuitively understood distances and sizes from the real environment. With translation gain, we may walk 1 meter in the real environment to move 2 meters in the virtual environment.

Figure 3.1: Translation Gain

Translation gain g_t is often expressed as the ratio of virtual displacement to real displacement [5]. If $S_{Virtual}$ and $S_{Physical}$ are the translation vectors for the virtual environment and the real environment respectively, then we can find the scale of translation between the two environments (see Equation 3.1). One of three scenarios are in play concerning the relation between $S_{Virtual}$ and $S_{Physical}$. The first is $S_{Virtual} > S_{Physical}$, when the resulting ratio is $g_t > 1.0$. This means that the user moves faster in the virtual environment than in the real environment. In the second scenario, the relation is $S_{Virtual} < S_{Physical}$ and the resulting ratio is $g_t < 1.0$. The user in this case experiences themselves moving slower in the virtual environment than in the real environment. The two variables can also be equal, $S_{Virtual} = S_{Physical}$, and the resulting ratio is $g_t = 1.0$. The speed at which the user is moving is then same in the two environments, meaning no translation gain is applied [5]. Regardless of the value of a specific relation, g_t indicates the scale at which we map movement from the real environment to the virtual environment.

$$g_t = \frac{S_{Virtual}}{S_{Physical}} \tag{3.1}$$

Although scaling user movement up/down may sound useful in increasing the virtual space that we can cover, it is not always beneficial. For example, Steinicke et al. [42] argue that humans in familiar environments, such as a realistic 3D city model, can accurately discern the difference between virtual and real translation movement. For this reason, the gains we receive from translation may be quite small in such scenarios.There are also certain concerns about users unknowingly changing their speed in order to compensate for the applied translation differences [49], but the displacement of scaling user movement could nevertheless prove beneficial in respect to the bigger picture. This would be similar to a cashiers asking "would you like to roll up your total?" This might be just a few pennies for a given individual purchase, but the total might accumulate into something worth noticing after numerous purchases. It is therefore our belief that translation gains would still be beneficial since the user could, more likely than not, remain in the virtual environment long enough for differences in translation to accumulate.

A study by Steinicke et al. [43] suggests that we can obtain translation gains that range from downscaling by a factor of 14% to upscaling by 26%, which would give us the range of [0.86, 1.26]. With translation gain alone, we would then be able to explore a 5.8 meter × 5.8 meter virtual environment in a 4.6 meter × 4.6 meter real environment [25].

3.1.2 Rotation Gain

We can also take advantage of the user's rotational movements, and there are multiple examples of when we can reorient the user in this way. For instance, the user may stand still and look around the virtual environment where they find themselves and seek their next task. The unique attribute of rotation gain g_r in such an example is that it can be applied when the user is standing still. Whatever the scenario may be, we can reorient the user by scaling their rotational movement in the real environment and mapping it to the virtual environment, and we can over-rotate or under-rotate the user's movements depending on the degree of scaling [27]. For instance, we could reorient the user to face the open end of the real environment so that we have more room to work with as the user moves around.

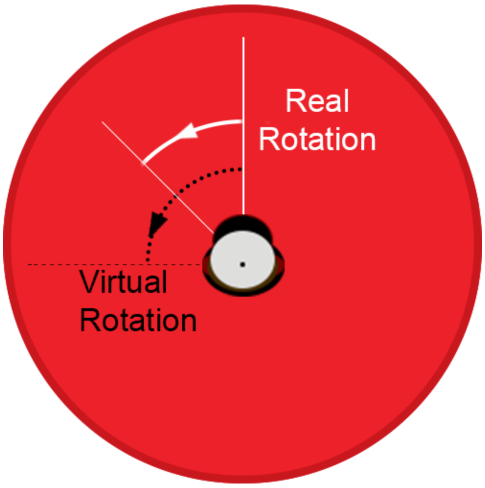


Figure 3.2: Rotation Gain

It is classically understood that there are three different elements that to define the rotation of an object in three dimensions(3D), namely, yaw,[2]

---

[2] The left/right movement that is used when we say "no."

pitch,[3] and roll.[4] Each indicates rotation around one of the axes of three-dimensional space. We can apply rotation gain to all three in theory, but manipulating yaw is most common [25, 45]. This is because it is more advantageous to manipulate rotation in the plane in which we are walking than otherwise. For example, we want to rotate a user towards the open end of the room. Imagine a two-dimensional, bird's-eye view game in which we are able to walk around, which is the plane we can walk in. We can see the top of the user's head from the bird's-eye perspective. Yaw is the movement that permits us to rotate the character around that point so that they face a different direction (see Figure 3.2).

Grechkin et al. [5] write that rotation gain is often defined as the ratio of virtual rotation to physical rotation. If $\theta_{Virtual}$ and $\theta_{Physical}$ are the rotational values for the virtual world and the physical world respectively, then we can find the scale in rotation between the two worlds. Equation 3.2 illustrates this ratio. One of three scenarios are in play when it comes to the relation between $\theta_{Virtual}$ and $\theta_{Physical}$. The first is when $\theta_{Virtual} > \theta_{Physical}$, and the resulting ratio is $g_r > 1.0$ in this scenario. This means that the user rotates faster in the virtual environment than in the real environment. Second, the relation could be $\theta_{Virtual} < \theta_{Physical}$, which gives us a ratio of $g_r < 1.0$. The user in this case would experience themselves rotating more slowly in the virtual environment than in the real environment. The two variables may also be equal, $\theta_{Virtual} = \theta_{Physical}$, which would result in a ratio of $g_r = 1.0$. The speed at which the user rotates is then the same in both environments, meaning no rotational gain is applied [5]. Regardless of the particular relation, g_r shows the scale at which we map the movements from the physical world to the virtual world.

$$g_r = \frac{\theta_{Virtual}}{\theta_{Physical}} \tag{3.2}$$

Since rotation gain is a static gain, with the user standing in one spot but rotating, we can estimate the size of the area that we can explore in virtual reality in comparison with what we have available to us in the real environment. In the Razzaque, Kohn, and Whitton [32] experiment, they were able to explore an approximately 8 meter × 20 meter virtual environment in their 4 meter × 10 meter real environment. This is a significant increase. The experiment was designed to have the user walk in the tracked space from one side of the lab to the other, turn, and walk back to the other side. This process was repeated until the study was completed and it could in theory be repeated an infinite number of times in order to explore an infinite area. However, the user had to be somewhat guided to stop and turn before they walked into danger. If a large prompt told the user every x distance that they now needed to turn, the immersion that virtual reality sought to accomplish would be broken. Rotation gain would therefore not quite work by itself, but it could still be quite beneficial.

Razzaque, Kohn, and Whitton [32] found that users did not notice

---

[3] The up/down movement that can be used to say ”yes.”

[4] The side-to-side movement that allows us to tilt our heads.

the additional rotational distortion they had injected. The vestibular stimulation added by injected rotational distortion became unnoticeable to the user because there already was a naturally high level of vestibular stimulation from their looking around [32]. The size of this initial ”natural” vestibular stimulation could thus have been the cause of the large rotation range that Steinicke et al. [42] discovered. Consequently, while the injected rotational distortion could be more easily detected if the user rotated slowly, there might be a lower risk of detection if they rotated quickly. We should therefore consider ways in which we can cause large vestibular stimulation. If we are able to do so, then we could at any point get the user to look around, which would give us larger rotational gains to work with [32]. This will be addressed in further detail in Section 3.2.

Razzaque [31] discovered in his early research that users were less likely to notice small injected rotations. He also demonstrated that a user is less likely to notice added rotations when their head is already turning. Since his initial findings, more research has been conducted to determine what the limits are before users notice injected rotations. Jerald et al. [11] write that users are less sensitive when the added rotations are in the same direction as head movements. In the same study, they found that users can be physically turned approximately 11.2% more and 5.2% less than the virtual rotation. Steinicke et al. [42] state that users tend to have problems sensing 90 degree rotation accurately, laying down the foundation of future research. Bruder et al. [2] in 2009 noted that we can physically rotate the user 30.9% more and 16.2% less than in virtual rotation. Steinicke et al. [43] would later observe that users can be physically rotated approximately 49% more and 20% less than in virtual rotation.

3.1.3 Curvature Gain

Curvature gain g_c extends Section 3.1.2. This redirected walking technique adds the component of movement to rotation. Razzaque, Kohn, and Whitton [32] found that users unwittingly rotate in the direction of the rotation, and that curvature gain while walking adds a seamless rotation to the virtual scene — these small rotations are perceived by the user as an error and they will correct their trajectory. The result is a real-world trajectory in an arc-shaped pattern while the user stays on their intended trajectory in the virtual world [5]. One example of the use of curvature gain concerns the user wanting to walk straight ahead for a long distance in the virtual environment. In this event, small rotations are added to the virtual scene, the user would observe them and they would correct their path to continue walking in a straight line in the virtual environment. By correcting themselves, however, they unknowingly walk an arc-shaped path in the real environment [42]. This is illustrated in Figure 3.3.

In a scenario where we want the user to walk on a circular path, but observe a straight path in virtual reality, the gains from curvature would be inversely proportional to the radius r of the curved path taken in the real world [5]. This ratio is shown in Equation 3.3. As the arc becomes tighter to the point that the radius is minimal, then the resulting curvature gain r

Figure 3.3: Curvature Gain

goes to zero ($r \to 0$) becomes infinite. Inversely, as the arc becomes larger to the point that there is no curvature, the radius becomes infinite. When r goes to infinite ($r \to \infty$), then the resulting curvature gain gives us 0. More simple stated, as r increases, the resulting curvature gain decreases, and conversely. Values greater than 0 would therefore correspond to increased curvature intensity on the walked path, while values equal to 0 would give none.

$$g_c = \frac{1}{r} \tag{3.3}$$

We can obtain curvature gains using different methods. Matsumoto et al. [20] use curvature gains to redirect their users through visuo-haptic interaction, which mixes the virtual environment and real environment together in order to merge visual and tactile perception. In their experiment, Matsumoto et al. [20] have a large circular wall in their lab with which users are in constant contact. In the virtual environment, the users are walking down a long straight path with a wall on their side. By mixing the two realities together, the user believes that they are in contact with the virtual wall, when it is in reality a physical wall. However, as discussed in Section 2.3, certain problems are associated with using constant touch, one of the most notable being the risk of walking into the physical wall if the walking trajectory in the virtual environment is changed. This makes it an undesirable solution. In addition, a labyrinth of walls must be pre-built to fit the virtual environment. This is not dynamic and not a desirable solution. Another way in which to use redirected walking involves visual manipulation. Seamlessly rotating the virtual scene causes the user to correct themselves, which causes change in real rotation, as in Kloos et al. [13] experiment. However, a good curvature algorithm requires some prior knowledge of where the user is headed in order to maximize the area available and keep the user safe [25]. Steer-to-center algorithms, which are based on always rotating the user towards the center of the room, are often used for this purpose [5]. Using this method does not require any prior knowledge of the user's path, which means that users could easily leave the tracked area if an unexpected move occurs. Combining steer-to-center with some form of warning that the user is leaving the tracked area could be a viable solution for our research. In either method, determining the area of the virtual environment that the user can now explore becomes unlimited if implemented correctly. If we assume that the user is walking in an endless straight path in the virtual environment, we can explore an infinitely large virtual area if the real environment is large enough for walking in a circle of an appropriate size.

A great deal of research has been conducted to find out how strongly we can rotate the user before they notice it. This has had varying results [25]. Steinicke et al. [43] suggest that we can get users to walk along the circumference of a circle with radius of 22 meters. Grechkin et al. [5] have since lowered the threshold to 11.6 meters or 6.4 meters. It has also been found that speed impacts the radius of the arc since users are less sensitive to noticing curvature gains at slower speeds [13]. These varying results may be due to methodological differences in the research, such as users' walking speed [5].

3.1.4 Bending Gain

Bending gain g_c refers to bending the user in a particular direction while they are walking along a curved path in the virtual environment. This is an extension of curvature gains, which we discussed in Section 3.1.3. The difficulty we are facing here is that the user can change their trajectory at any point. This could be problematic in a confined space since the user might walk directly into obstacles. The issue with bending gains then becomes the need for prior knowledge of the user's path or a good prediction of where they are headed. Moreover, immersion issues might arise if we do not have a good prediction — if we incorrectly apply bending gains, the user might detect the applied injections [25]. This could also cause motion sickness and ruin redirected walking.

$$g_b = \frac{r_{Virtual}}{r_{Physical}} \tag{3.4}$$

Langbehn et al. [18] used the formula in Equation 3.4 to determine how strongly we can bend the curve being walked. If $r_{Virtual}$ and $r_{Physical}$ are the radii of the walked paths in the virtual environment and the physical environment respectively, then we can find the scale in radius between the two worlds. One of three scenarios are in play when it comes to the relation between $r_{Virtual}$ and $r_{Physical}$. In the first, $r_{Virtual} > r_{Physical}$, which results in the ratio $g_b > 1.0$. This means that the user walks in a larger arc in the virtual environment than in the real environment. In the second, $r_{Virtual} < r_{Physical}$, which results in the ratio $g_b < 1.0$. The user in this case would experience themselves walking a smaller arc in the virtual environment than in the real environment. Finally, the two variables may be equal, $r_{Virtual} = r_{Physical}$, which results in the ratio $g_b = 1.0$.

The radii of the arcs in which the user walks are then the same in both environments, meaning no bending gain is applied [5]. Regardless of the specific relation, g_b indicates the scale with which we map the movements in the real environment to the virtual environment.

A good bending gain algorithm can be quite useful for expanding the area that we can explore. Langbehn et al. [17, 19] suggest that when users are constrained to a specific path, it is possible to explore a 25 meter × 25 meter virtual environment in a 4 meter × 4 meter real room [25].

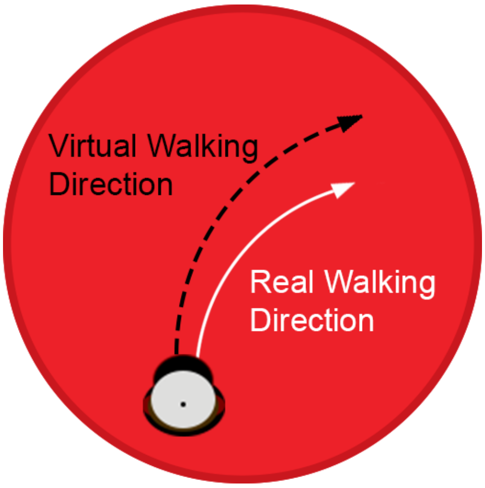


Figure 3.4: Bending Gain

Research on bending gains is quite new, and thresholds for when a user detects the injected changes are still debated. Langbehn et al. [19] argue, however, that we can bend the user's virtual path by a factor of 4.4 in respect to their real path.

3.2 Attention Grabs

Redirected walking is currently not fool proof — it can at times break and even cause users harm. This may be the case because certain people have an easier time than others at walking in a straight line when they lose sensory inputs. It might be more difficult to redirect them in such scenarios since they are more sensitive to where they are headed. It might also be the case that the user made an unpredictable movement that threw off the algorithm. In such incidents, we need to ensure that we can safely stop the user before something negative happens. We then need to be more concerned with stopping the user from getting into harm's way than keeping them immersed.

Imagine yourself walking down the street in a familiar place. You might be looking around if it is a nice day, or you might be concentrated on making sure that you are not walking astray. You suddenly shift your focus onto one specific thing. It might be a bird flying by or the loud high-pitched scream of a child that dropped its pacifier. In any case, you now find yourself focused on that object. These are examples of how attention grabs work — an attention grab is something that captures a person's attention. It could be subtle something as subtle as a speck of dust floating in the air or something more ”in your face” such as a floating stop sign in a video game. In theory, anything could serve as an attention grab. The bare minimum requirement is that it is something that catches the user's attention. This includes other forms of attention such as focusing on the ball in tennis, reading a book, the road/path taken to reach your destination, etc. However, research into redirected walking continues to focus on one specific element at a time. To the best of our knowledge, only a very limited amount of research has addressed an open world in which the test subject is asked to walk freely. As a result, little research has been conducted regarding the effects of attention grabs in free motion.

Currently, most research regarding redirected walking has addressed guided walking, not walking freely, even though an ideal redirected walking algorithm would take such freedom into account. The methodology of a guided walking test could involve having the test subject perform a certain set of tasks within a suitable virtual environment with limited points-of-interest. A point-of-interest in this instance would be a location the user would like to get to. Razzaque, Kohn, and Whitton [32] instructed their test subjects to perform a simulated fire drill in which four buttons mounted on the virtual walls that served as waypoints were to be pushed in order. Grechkin et al. [5] asked their test subjects to walk in the direction of a dot. Since there was little else to do in these virtual scenarios, these tasks become the lone points-of-interest and attracted the full attention of the test subject. This made it possible for researchers to take full advantage of the physical space to which they had access because the paths that the test subjects could take became predictable — test subjects would not stray off the predicted virtual and real paths. However, if we would wish redirected walking to become a generic application — something that could be added to any virtual reality application — then the algorithm must take into account unpredictable virtual paths with an uncertain number of points-of-interest. In large open world environments, such as Minecraft, the user can at any point change their point-of-interest and choose to walk in a different path, which makes where they want to walk unpredictable. A larger buffer is thus required to permit such unpredictable movements to occur. This differs from the situation presented by Razzaque, Kohn, and Whitton [32], where the test subject walks close to a wall before they reach their objective because the virtual environment was designed to fit the real environment.

In order to achieve full redirected walking in virtual reality, tracking potential points-of-interest — such as villages, trees, caves, etc., in Minecraft — in order to allow for a buffer large enough for likely changes in walking to occur becomes a factor. This still could lead to users walking into walls if the buffer was not large enough. Last resort measures should be in place to quickly reorient the user safely. Through the use of attention grabs — something that grabs the attention of the user — we can distract the user from their point-of-interest long enough to make a drastic change in their real path. In the light of our discussion in Chapter 2, we will focus only on attention grabs through the use of audio and visual stimuli. Each type of attention grab is further discussed in its own subsection:

- Visual grabs (Section 3.2.1)
- Audio grabs (Section 3.2.2)

This section briefly explains these two general forms of attention grab. We also propose that there are two more specific types of attention grabs, namely, *guiding*, and *obtaining*. The former presents a way for guiding the user through the virtual environment. A pertinent example to visualize the way in which the user should walk is as the dot used by Grechkin et al. [5]. The latter, which is mainly used to get the user out of danger by abruptly shifting their attention, is often connected with repositioning the user by means of rotational gains. This was described above in Section 3.1.2. We will here discuss the use of guiding and obtaining attention grabs for the two general forms mentioned above.

3.2.1 Visual grabs

Briefly stated, a visual grab is something that grabs the user's attention by stimulating the visual system. This means getting the user's attention through the use of visual objects that they can see in the virtual environment. The advantage of using visually based attention grabs is that virtual reality is a visual-based system. In order to appeal to the user's visual system, we would thus only have to create an event that would shift the user's focus on the basis of visual events. This could be something as small as a butterfly guiding you to your objective, a large fence blocking your path, a visible timer that requests you to go back into the playable area, or anything else that uses visual cues to gain the users attention.

A visual based guided attention grab is useful for redirected walking techniques that utilize movements. This would include all redirected walking techniques except for rotational gains. We mentioned in Section 3.1.4 the issue of needing a good prediction algorithm to make this redirected technique work. Visual attention grabs might help solve this issue. For example, in numerous video games, users can place waypoints on a map to figure out the general direction in which they would like to go — they tell the user where their destination is. Different video games implement such waypoints in different ways, but generally speaking, all have some way for visualizing where the user should go. Some games even find the quickest path the user can take to get to their destination and presents it. This acts analogously to how a person would enter an address in their car's GPS system in order to understand how to drive to their destination. When the user themselves places their waypoint, we can assume that they will follow the path indicated to them. It would then become easier to predict where the user will walk, and we could then utilize both curvature gains and bending gains to their full potential. For instance, if we were able to obtain this pathing, or even create our own when the game fails to make one, then we could map the intended path from virtual space to physical space. If we assume that the findings presented in Langbehn et al. [17, 19] are correct, then combining them with the use of guided attention grabs in the form of waypoints could be quite beneficial. Other forms of guided attention grabs can be more subtle. Grechkin et al. [5] use a dot to guide the user, and the user focuses so intently on this dot that when the virtual scene rotates, the user corrects themselves unknowingly.

The redirected algorithm may fail in some instances, leading the user to walk into danger. We have seen in Section 3.1.2 that we can obtain large gains even when the user stands still. If we succeeded in stopping the user and having them look around, then we could utilize rotational gains to reorient them to face the open end of the room. We can use visual based obtaining attention grabs to do so. A buffer zone should also be looked into in the effort to stop the user safely without breaking their immersion. One interesting method for this purpose discovered by Kloos et al. [13] involves human interactions. Kloos et al. [13] built upon a significant amount of research about human interaction when being in another person's interpersonal space. Since they suggest that humans tend to steer away from other humans' paths, they successfully used other avatars as a means for reorienting the user in their research. Although this feels like a natural way to reorient the user, it may not work in all situations where we need the user to diverge from their intended path. If the user is placed in a city simulator similar that which Kloos et al. [13] built, then this reorientation technique using a visual grab seems fair. However, it may not be the best natural approach to have human avatars encroaching on a user's personal space in order to redirect them in a first-person shooter or a building game. One alternative would be to have game developers — or other uses of virtual reality — implement their own means of reorientation. An example of this can be seen in Chen and Fuchs [4], who found great success with redirection when the user came close to crossing a boundary. In their experiment, they created a video-game-like virtual environment in which a dragon flew by the user when they were close to the edge. Nonetheless, generic solutions should still be explored in the event that a video-game company does not invest in proper attention grabs.

We also need a fallback plan if neither of the efforts above work. One possibility would be to break the user's immersion and let them know that they have to stop. If this was done, then the user should also know what they need to look for. Otherwise, they would not know what the sign of danger looked like and could be harmed. Kloos et al. [13] used a big stop sign when this was the case. Another way in which to convey such information would be to use the designated setting in Oculus' Guardian — when the user nears the boundary of their tracked area, a grid-like wall appears to let them know they are close to the edge. If the user passes through this wall, the headset switches from the virtual environment to a transparency mode that shows the real environment. The user then sees where they are and can reposition themselves accordingly.

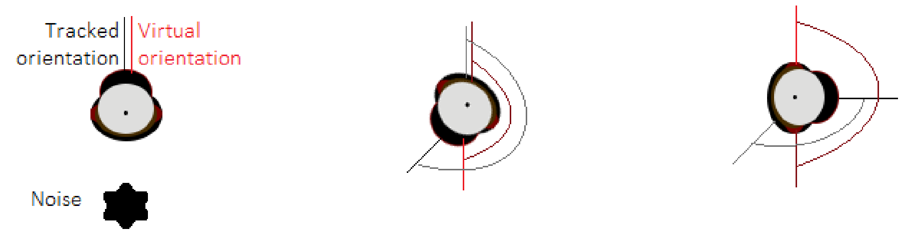
3.2.2 Audio grabs

Most simply stated, an audio grab is something that grabs the user's attention by stimulating the auditory system. Although the use of audio has often been overlooked in research concerning redirected walking, it has been proven to be quite useful. Razzaque, Kohn, and Whitton [32] suggest that 3D spatial sound is an important factor in being fully immersed because the brain can detect small discrepancies between auditory and visual cues. A consistency between these two systems would thus make it more likely that a user would interpret rotation, for example, as self-motion [32]. However, the sound must match the environment — if you were playing a farming simulator, having a bomb explode as your audio grab would not fit in and would thus break the immersion that we were trying to protect.

Audio grabs could also provide a means for guiding the user to a destination, as discussed under Section 2.2. Humans rely upon other senses, such as audio, when their sense of sight is compromised. We could in theory manipulate audio stimuli to make the user walk on a circular path, but perceive that they were walking in a straight line, if we moved the sound source accordingly. This is one form of an audio-based guiding attention grab. Nilsson et al. [26] also explore whether we can achieve even further gains with the use of spatialized audio. Although it is not conclusively proven, they believe that the use of audio could have an effect in situations where it is harder to see, such as under foggy or dimly lit conditions [25, 26]. This would make sense since if there are no visual cues that the user is in danger, they must rely on auditory cues to determine what to do. Without a sound source, however, the user would not have a reference point for where they were and could well get lost. Examples of audio-based guiding grabs would thus involve continuous sounds that have a reference point, such as a campfire, a running car, or music playing over speakers in a grocery store. The issue with using sound as a guide is that sound has a very limited range. A campfire would only be heard for a few feet, and a running car would only increase that range by a small margin. Unlike what we saw in Section 3.2.1, we do not have enough research to conclusively prove its value as a form to guide the user.

In our opinion, a better use of an audio grab is an audio-based obtaining grab. As mentioned above, sound provides only a limited range to work with, which makes it difficult to guide the user. However, we could adjust the intensity of the sound to make it stand out — the user might react to a sound that stands out. For example, hearing a loud noise behind the user could cause them to quickly turn around to investigate what it was. If this happened, then we would have a chance to reorient the user. We could over-rotate the user in the real environment when they turned to investigate, then under-rotate them when they turned back towards their initial position to continue their journey. This would cause a change in the user's orientation in the real environment that we could utilize. This process is illustrated in Figure 3.5.

If none of these approaches work, then we should consider "last resort"



(a) User's initial rotation in the virtual and real environments

(b) User's rotation after investigating sound

(c) User's final rotation after rotating back to initial virtual rotation

Figure 3.5: Timeline of user investigating a new sound

efforts to keep the user out of harm's way. At this point, it would no longer be important for us to consider keeping the user immersed in virtual reality since the situation called for breaking immersion for safety reasons. Razzaque, Kohn, and Whitton [32] solved this by simply telling the user to stop walking, followed by requesting the user to look left and right. Doing this gave their algorithm a chance to correct itself and redirect the user to a safer area. By doing so, they discovered that it is possible to reorient the user more than 90 degrees without them noticing the injected rotation.

3.3 Chapter Summary

Reorientation Techniques In this section, we discussed the various techniques used to manipulate the mapping between the real and the virtual environments.

Translation Gain is used to change the user's walking speed. This is quite easy to implement and we have employed it in our study.

Rotation Gain requires no prior knowledge of the user's intended path. Since the virtual reality template in our game engine include a way to rotate the user, implementation was quite easy. We have also employed this in our research.

Curvature Gain requires prior knowledge and can be tricky to implement. However, we can to a certain degree force the user to walk a specific path in a testing environment. Implementation is thus not an issue. Since we believe curvature gains possess a great potential, we have looked into them further in our research.

Bending Gain is an evolution of curvature gain that adds the ability to further bend the user if they walk on a curved path. Since testing every gain takes a great deal of time, we decided to not use it in our research at this time.

Attention Grabs This section focuses on the various techniques used to gain the user's attention through visual cues and audio cues.

Visual Grabs emphasize visual elements that stimulate our visual system. We are interested in testing the effects of such grabs and have implemented them in our study.

Audio Grabs emphasize using auditory cues for gaining the user's focus. While we have not used this to the extent discussed in this subsection, we have used audio cues that let the user know when a visual stimulus has been activated. We also chose to not use sound-cancelling headphones. This was for the sake of safety since we can vocalize to the user when they are in danger if other safety measurements do not work.

Chapter 4

Study Design

Before any lines of codes can be written in a research project, a high-level preparation to the project should be in place. This is a stage in the planning phase of a project. The reason we devoted a significant period of time to planning our project was to limit problems that might arise down the road because of improper planning. For example, a common practice in business is to have daily meetings at which people discuss what has been done, what is in the works, and what needs to be done in order to ensure that all members of the team are on the same page so that projects proceed smoothly and correctly. In our case, improper planning could lead to not having data that would be important during the analytical phase of our project, or wasting time because of having selected improper tools, and many other problems as well. In this chapter we present the theoretical approach we adopted in order for our project to work and the various methods we decided to use.

Developing a theoretical approach to research involves taking multiple items into consideration. Our main concerns were the research objectives, how to collect data during our research, which research procedures to use, and the various tools and analytical methods needed to obtain sound results. With this in mind, we divided this chapter into the following sections:

- Research Objectives (Section 4.1)
- Procedure (Section 4.2)
- Type of Design (Section 4.3)
- Sampling Method (Section 4.4)
- Data Collection (Section 4.5)
- Tools and Analysis Methods (Section 4.6)

Shortcomings in the preparation phase of research can lead to needless difficulties. The goal of this chapter is to present the various items that we took into consideration to ensure that our project would meet its goals.

4.1 Research Objectives

In this thesis, we are interested in finding out how much we can redirect the user before there is a negative impact on their quality of experience in virtual reality. This is based on the knowledge presented in both Chapter 2 and Chapter 3. We also presented our research questions in Section 1.2. Against this background, we may present our research objectives as follows:

Research Objective #1 To evaluate the usability of redirected walking systems. For this purpose, we focus on (1) how strongly users perceive redirection and (2) their overall quality of experience.

Research Objective #2 To understand the perceptual threshold of the amount of redirection that causes the user to experience distraction,[1] loss of satisfaction,[2] or a negative physical effect,[3] such as discomfort.

Research Objective #3 To investigate the use of distractors in combination with redirected walking. We will focus on whether the perceptual threshold changes when distractors are present in the virtual environment.

Research Objective #4 To explore how different virtual environments affect the perceptual threshold of redirected walking.

4.2 Procedure

Our third research objective (see Section 4.1) is to investigate the use of distractors in combination with redirected walking. This implies that we need to explore a given redirected walking technique for both with distractors and without distractors in the same virtual environment. We limit the number of redirected walking methods to explore this objective down to rotation gains for reasons discussed in Section 3.2.1. Our fourth research objective states that we want to explore how different virtual environments affect the perceptual threshold of redirected walking. This implies that the same redirected walking method is to be tested over different virtual environments. Our distractor research objective fits well to be one of these virtual environments. The second virtual environment must therefore be used for testing rotational gains. Our first and second research objective, however, includes both curvature gain and translation gain. These gains are not fit for distractors and thus it should be tested without them. To limit the number of tests needed to complete our research, the second rotation gain's virtual environment could also be tested without distractors, and be grouped with the remaining redirected walking techniques.

---

[1] Despite noticing the applied redirection, users do not find it to affect game play.

[2] The applied redirection is noticeable and negatively impacts the game experience.

[3] The applied redirection causes the users to show symptoms of virtual reality sickness.

For this reason, we plan to conduct two separate studies for this thesis, namely, the base test and the distractor test. The base test focused on finding the detection threshold of various gains in a sanitary virtual environment. It included testing for translation gain, curvature gain, and rotation gain, each of which was associated with a given virtual world (see Table 5.2) and tested over different levels of gain (see Table 5.1). The distractor test focused on comparing how users found redirected walking when distractors were introduced. With this aim, we compared how users found redirected walking with and without distractors over the same level of gain within the same virtual environment. We only used rotation gain in this test since we viewed it as the best fit for the use of distractors (see Section 3.2.1). Users were also introduced to a more game-like virtual environment, the dungeon. We could therefore further analyze the effect that the virtual environment has on redirected walking, comparing a game-like environment (dungeon) to a sanitized environment (fire drill). The user was put through a sequence of tests in both of these studies. This was done to test the different levels of gain against each other in order to assess how noticeable the different algorithms became in their given environment.

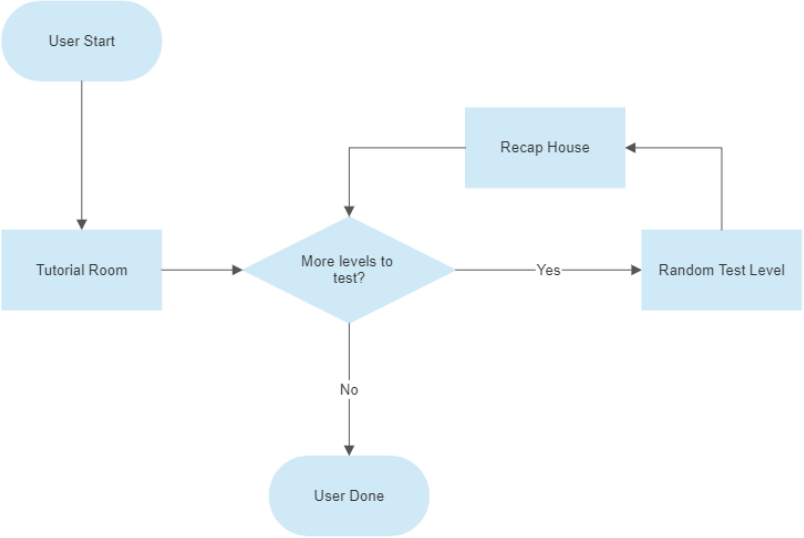


Figure 4.1: Test progression

Figure 4.1 illustrates the process users followed in our study. When a user began the test, they were placed in a tutorial room. From there, the algorithm entered a loop that ended when there were no more variables to test. When the loop ended, the test was over and the user exited the virtual environment. The loop involved selecting a scenario from a series of scenarios that had not yet been tested, into which the user would be placed. Once a given test was completed, the user would be placed at a virtual intermission level for discussion, questioning (see Section 4.5) and resetting. Once this was completed, the user returned to the start of the loop to be placed into another scenario.

4.3 Type of Design

Virtual reality in its current state is used by a wide range of people across the world. Redirected walking as an extension of virtual reality is thus used by humans across the spectrum. We saw it as necessary to conduct a study using human volunteers in order to properly test the properties of redirected walking, and we viewed designing our research as a user study to be the best approach.

By means of a user study, we were able to obtain information directly from the users of virtual reality about their experiences. Each person experiences and observes an event differently from another. We might thus see instances in our data set where some users reported a high quality of experience while others reported low quality of experience. There can be multiple reasons for why this happens. However, since we believe that virtual reality is something that everyone can enjoy, our redirected walking algorithm should fit the general population. We therefore turned a blind eye towards why users might give the answers they gave and analyzed the entirety of our population.

We looked at replication and partition in order to ″normalize″ the data provided by the participants in our user study. Replication and partition were used to analyze data by distinguishing between what we may categorize as ″same″ or ″different″ based on our defined criteria [21]. Most research into redirected walking has focused on when the algorithm becomes noticeable. We believe, however, that there should be an added factor for whether detected redirection negatively impacts the virtual reality experience. Our partition groups were divided into three subgroups based on their experience with redirected walking. A common practice is to use the Likert scale — an attempt to assign values to discrete labels in such a way that it makes sense to give distance between neighbouring terms equal weight. We chose to follow this practice, by using a scale from 1 to 5, and assigned the values seen in Table 4.1. These groups were defined for the user as guidelines during questioning.

Value	Description
5-4	It does not impact the experience of virtual reality
3-2	It negatively impacts the virtual reality experience
1	It causes users to show symptoms of virtual reality sickness

Table 4.1: Partition groups for determining quality of experience

Issues may arise during development in any programming project. We decided to use TRIZ to solve any problems we might potentially encounter. This method is based on the general idea that somebody, somewhere has already solved any problem you might be stuck with, or something similar [34, 39]. This would mean that we could find prior research or solutions that would help us with what we were doing in our project.

4.4 Sampling Method

The participants in our study were selected through convenience sampling, which means that we sampled a group of people that were easy to contact [33]. Since our research group consists of students from the University of Oslo, we reached out to other students (through verbal communication, posters displayed at various locations at the institution, etc.) and to other organizations that we are associated with. The only requirement we had for performing our tests was that the participant be able to walk and simultaneously interact with the virtual environment, regardless of age, gender, previous experience with virtual reality, etc. Because of the limited time available for a master's research project, our goal was to have between 15 and 20 participants — a group large enough to obtain statistically significant results — for each of our tests.

During the selection process, we provided potential participants with our problem description and a basic definition of redirected walking. However, no information was provided about the different algorithms and when they would be used to limit any possible bias. If a potential participant was interested, they signed up for an available time slot. Participants were also given an information sheet prior to their tests. This included information about what to do at the various test levels, the kinds of questions we would ask, and the different measures we used to keep participants safe with in the virtual environment. This information was provided in order to prepare the user so they knew what to do during our tests, which limited the amount of time they were not active during testing. Since we were not testing the user's ability to complete a test, but rather how they interacted with the virtual environment, this would not affect any of our study's findings.

4.5 Data Collection

Each participant was asked to fill out a basic form when they registered for our research. We asked participants their age, gender, height, experience with virtual reality and gaming, sense of balance, and whether they used anything to correct their vision. The aim of these data points was to see whether there were differences within the population based on their attributes.

Following the completion of each separate test, participants were asked a series of questions orally while still in the virtual environment. We wanted them to reflect on their experiences as soon as possible. Oral responses are also more time-efficient than filling in forms in the virtual environment and less disorienting than leaving virtual reality to answer a questionnaire. The three questions used for our study may be seen in Table 4.2. Question 1 is a ”Yes” (1) or ”No” (0) question asking whether participants noticed any reorientation effect. If a participant responded that they had noticed unnatural movement, we proceeded to Questions 2 and 3, which explored participants' subjective experiences and are based on the psychometric scale from 1 to 5 defined in Section 4.3. If a participant responded to Question 1 by indicating that they did not notice any reorientations, we assumed that they had no issues with the experience (5) and had no discomfort (1) because they regarded the test as a normal virtual reality experience. Participants also had the opportunity to add any relevant information they wished. Finally, we asked participants follow-up questions when they were done with all of the tests. These questions were inspired by the game experience questionnaire, and we used them to gain a better understanding of how immersed the participants were in the virtual reality experience. These questions were mainly for use in future work in the hope of creating better virtual experiences for participants.

Question	Question description
1	Did you notice the applied reorientation?
2	How good was your experience?
3	How much discomfort did you feel?

Table 4.2: List of questions asked during the study

Redirected walking was used to explore a larger area in the virtual environment than what was available in the real environment. This is done by manipulating users' movements in the real environment before sending them to the virtual environment. We therefore wished to be able to compare differences in movements between the two environments. The easiest way of doing so was to compare the paths walked in the two environments to each other, which meant that we needed to extract users' coordinates (position) in both the virtual environment and the real environment. This allowed us to analyze their movements and see whether they were able to explore a larger area than what was provided to them.

Knowing only the user's position would be quite abstract, however. All we would then have access to would be the coordinates for their positions in the virtual and real environments. Comparing the paths a user took in those two environment would enable us to make assumptions about how the user was redirected, but the coordinates by themselves would not tell us the difference in rotation between the points. Rotational gains and curvature gains emphasize the user's change in orientation between the two environments. It was therefore beneficial to gather data about the user's orientation in the two environments to see, for example, how much we could change the user's rotation within a certain interval. The main drawback of gathering data orally is that the questionnaire had to be linked to the data collected from the game engine. To do so, we additionally stored the order in which the scaling and redirected walking took place. In post testing, we used this data to link the answers from the questionnaire to a specific test that the user performed.

4.6 Tools and Analysis Methods

There are several factors we had to consider when selecting the types of tools we wanted to use for our analysis. The main factors worth mentioning are the handling of large data sets, the visualization of raw data, computing speed, graphing capabilities, and statistical analysis capabilities. There are numerous of data analysis tools to choose from, namely, Microsoft Excel, Python, R, SPSS, MatLab, etc. We will mainly focus on Excel and Python due to lack of experience with the other tools.

Microsoft Excel has long been a popular choice for data analysis. Python has risen in popularity for data analysis, however, thanks to the *matplotlib* and *Pandas* libraries, which we can use to create our own software for graphs (matplotlib) and analyses (Pandas). The use of Python is often preferred for analyzing large data sets since it is much faster than Excel. In addition, it includes graphs (such as the violin plot) that currently cannot be created with Excel. A major drawback is that it takes time to learn and setup the code. Excel, on the other hand, has the ability to act as a database for visualizing all of the raw data for a given user in one place. Being able to see the raw data in this way can be beneficial since we can visually compare the different responses and find outliers more quickly than by writing an algorithm. The Excel interface also allows for quicker modification of data, such as transient removal, removing outliers, change graph design, etc., than if we were to modify code for this purpose. The issue with Excel comes down to its ability to process large data sets. Since we did not envisage our studies becoming large enough to cause such problems, we used Microsoft Excel as our data analysis tool.

We performed a statistical analysis of our data in order to determine their statistical significance [9]. This was done by hypothesis testing. In order to use this method of analysis, we needed to define a null hypothesis (what is assumed to be true) and an alternative hypothesis (true if statistically significant). We defined these hypotheses as follows:

Null Hypothesis There is no significant difference between the two.

Alternative Hypothesis There is a significant difference between the two.

We used Analysis of Variance (often referred to as ANOVA) to compare the statistical significance of different groups [36]. It should be noted that ANOVA can only be used on the assumption that a given participant is included in each defined group. In other words, we cannot use ANOVA to compare how men, women, and other genders responded since the participants cannot belong to more than one such group. To resolve this issue, we used Chi-Squared test (χ^2) [22] when we could not use ANOVA. One such example involves comparing whether our sample properly represented the population.

Chapter 5

Experimental Setup

In all research, the goal is either exploration or to find evidence that either supports or rejects a hypothesis. For this purpose, there must be a way to perform such experiments. In our case, we conducted research to see how much we can manipulate a user with various redirected walking methods before that negatively impacted their experience in virtual reality. This chapter is devoted to describing our experimental setup and providing information regarding the configuration of the redirected walking methods used in our study.

There are numerous items that needed to be taken into consideration for our experimental setup. This included game engine, virtual reality goggles, operating systems, implementation of redirected walking methods, and other elements. These may be categorized into two bins: one concerns the process of selecting the equipment used for this study, while the other concerns how we used this equipment to create the software for performing the reorientation. Virtual environments must also be created for performing our tests. Finally, a pilot study needed to be conducted to ensure that everything worked as it should before moving on to the real tests. This chapter contains separate sections for each of these categories.

- Tools and Platforms (Section 5.1)
- Algorithm (Section 5.2)
- Virtual Environments (Section 5.3)
- Pilot Study (Section 5.4)

Our goal with this chapter is to provide information and reasoning regarding why we chose the various configurations and how we implemented the different algorithms used in our research. We hope that our research will be built upon in the years to come, and our aim is to make our research process as transparent as possible so that it can be replicated. We would be pleased if future master's students and other researchers were inspired by our approach to proceed further on the basis of our work.

5.1 Tools and Platforms

Selecting functional tools and a platform is essential in any given study. In our study, using a computer with limited processing power might affect our tests by limiting performance in terms of smoothness (number of frames per second), resolution, and other respects as well. In this section we will introduce the various tools and the platform that we used in conducting our study.

5.1.1 Game Engine

Briefly stated, a virtual environment is the world that the user experiences through the use of virtual reality goggles. This research is based on how we can fully immerse the user into virtual reality, and a key issue is to create an environment into which we can place our test subjects. Some thought is therefore needed to determine how we would like to proceed. Our main objective is to redirect the user in a way that allows them to explore an infinitely large area in virtual reality while remaining in a finite space in the real world. Designing our virtual environment must then take into account how we can create virtual environments that are larger than the physical space to which we have access.

We had two options for finding a way in which for implementing these virtual environments:

1. Develop a system ”in house”.
2. Use a third-party game engine as our baseline.

Both of these options have positives and negatives. Developing something ”in house” means that we would take the time to create the system that we needed by ourselves rather than use a third-party application to solve the issue for us. Creating a virtual environment from scratch would grant us the creative freedom to do whatever we would like. We could implement our own physics engine, determine how objects would interact, implement redirected walking directly into the engine, etc., without going through the layers of abstraction that a game engine would require. On the other hand, developing something from scratch is quite time consuming, and using a third-party engine would resolve the issue of time constraints. The system would already be created for us, and all we would need to do is add our redirected walking algorithm and create virtual environments to test it. However, we would also lose a degree of creative freedom since our work in this respect would be restricted to accessing and modifying the game engine. Nonetheless, little time was available to us to develop a system as advanced as an already-existing product. For this reason, we found it more advantageous to use a third-party game engine.

There are two competing game engines to consider on the open market: Unity and Unreal Engine. There are other options beside these. The majority of them, however, were either expensive or lower quality. We therefore do not consider them for our research. Both Unreal Engine and

Unreal were valid options for our research since they allow us to implement redirected walking and create virtual environments. One of the larger differences between the two (with implementation in mind) involves the coding language they use to create their game engine — Unreal Engine uses C++ while Unity uses C#. Another difference concerns how accessible the game engine is for any needed modifications. Unity is more restricted than Unreal Engine in this respect, which could cause problems down the road. Unreal Engine is open-source, however, and this makes it possible to manipulate the game engine if needed. We consequently agreed on using Unreal Engine 5, the latest version of Unreal Engine, for our research.

5.1.2 Virtual Reality Headset

Virtual reality goggles are a necessary tool for this thesis for multiple reasons. For example, it disconnects the user from the physical surroundings, blocks the user's view to the real environment, and displays a virtual environment to the user that they can explore. There are currently numerous options for this type of equipment on the open market, including Oculus Quest, Oculus Rift, and Vive Pro. For our research, we were mainly looking for a virtual reality headset that would permit us to traverse a large open space (approximately 8 meters × 8 meters).

One major issue with modern day virtual reality headsets is that they are often wired (or tethered) to the gaming system, as is Oculus Rift. This means that the user's range of motion is limited to the length of the cable that connects the virtual reality headset to the gaming system. This imposes a significant risk since the user might walk farther than the length of a shorter cable and damage the entire gaming system by pulling it onto the floor. Users might then reject the idea of physical movement if it could result in financial losses. Furthermore, it was noted in Section 2.3 that feeling the drag of a cable from a tethered headset could break immersion for the user [42]. A static solution might therefore be more desirable.

A static solution involves the user being placed in a location in the real environment and moving within the virtual environment by means of a different system, including such traditional gaming devices as a joystick, mouse, and keyboard. A static solution does not necessarily mean that the user would be standing still, however, since treadmills have already made it possible to be in motion without changing your location. Progress has been made in the use of treadmills for virtual reality. Unlike the typical treadmills found at the gym, where you can only go in one direction, virtual reality treadmills allow a full 360 degree range of motion. These types of devices have a high price tag, however, and also feel unnatural to the user [32]. Using a static solution is thus not beneficial for our research.

One possibility for overcoming the limitations of tethered virtual reality headsets and static systems would be to use wireless virtual reality headsets. A number of models are available on the market, such as the Oculus Quest series. Using this type of virtual reality headset could prove beneficial since there are no cords that might break immersion or comprise a hazard for the user. They still have a maximum range of use, but the negative impact of exceeding that range would not be as significant as with their tethered counterparts. However, wireless virtual reality headsets lack the power, and thus the graphics, that tethered virtual reality headsets have, and this could be problematic. We wanted the participants in our study to have a good experience when case testing. Less power could mean that there would be a latency that participants notice, and this could lead to test results being skewed.

Between the two options regarding virtual reality headsets, finding a solution that increases the range of a tethered system, such as a laptop in a backpack, is problematic for various reasons, such as the limited computing power of laptops. Although there is a latency difference between the two options, it is more or less insignificant to our research. Since we were primarily interested in a solution that gives us a wider range, we settled on using Oculus Quest virtual reality goggles in our research.

5.1.3 Hardware

Early in our research we used an older computer for development. We noticed, however, that it lacked the necessary computing power to run a smooth virtual reality experience. We therefore switched to a newer computer and observed a significant increase in smoothness. The new computer was equipped with a NVidia RTX 4080 Graphics card and a Ryzen 9 7800X CPU, which ensured ample processing power and the ability to deliver a virtual reality experience with no bottlenecks.

One thing to consider was how to connect Oculus Quest to the virtual environment. We launched virtual reality from our computer to ease the logging of data, which meant that we needed to connect the headset to the computer. We used Oculus AirLink for this purposed, with the connection being served over a 5GHz WiFi link using a wireless router.

5.2 Algorithm

There were several points to consider when developing our redirected walking system. The two main examples were deciding how to implement redirection and providing safety measures. This section provides information about how we created a redirected walking system that included smoothing elements to hopefully reduce detection and safety features such as resetting. It should be noted that we used *OpenXR* in preparing for the use of a wide range of virtual reality headsets in our research. OpenXR is an open standard widely supported by manufacturers and development suites.

5.2.1 Modifying Unreal's Motion Control Input

Our initial assessment of the Unreal Engine 5 virtual reality suite revealed that the default manner in which it processes virtual reality input required additional work on our part because any virtual reality goggles control input applies a relative offset to the camera attached on the avatar, not to the avatar itself. Consequently, a transform must be calculated in order to apply an offset rotation to the camera so that the avatar pivots around the camera in an orbit. We modified the OpenXR function that reports the location of the virtual reality goggles to Unreal Engine 5 (`GetCurrentPose()`) to facilitate that operation. By adding a callback function into our code and returning a zero-vector for the location, we eliminated any relative offsets added by Unreal Engine 5 since it believed the virtual reality goggles had not moved. The actual location of the virtual reality goggles was then applied to the avatar manually through our callback. The results of this change were that the camera was directly coupled to the avatar and, consequently, any rotation applied to the avatar was translated directly onto the camera.

5.2.2 Calculating Redirected Walking Methods

The user was represented in the virtual environment as a pawn. A pawn does not move — the viewport[1] associated with it instead floats independently, representing the user's movements, as an offset to the pawn. As mentioned in Section 5.2.1, we made a slight change that makes the pawn and viewport float together in the virtual environment. However, this still maps the user's movement between the two environments as a 1:1 ratio — whatever the user does in the real environment is replicated in the virtual environment. We must break this 1:1 ratio to achieve redirection. In order to do so, we intercept the changes taking place in the real world that occurred between frames and manipulate them before applying it to the virtual world. Since the viewport acts as an offset, we implemented our redirected walking by applying the redirection to the pawn itself, not to the viewport. For example, if the user is exposed to a rotation gain of 1.5, the user's camera rotates by a factor of 1 (untouched) while the pawn rotates in the same direction with a factor of 0.5.

Technique	Levels				
Distractor Rotation	1.0*	1.1	1.25	1.5	
Base Rotation	1.0*	1.1	1.25	1.4	1.55
Translation	1.0*	1.1	1.15	1.2	1.25
Curvature	∞*	12.5	10	7.5	5

* null settings

Table 5.1: Overview of the levels under test

Table 5.1 presents the different conditions that correspond to varying levels of translation, curvature, and rotation (for distractor test and base test). We use the *null setting* as the baseline, which indicates that no redirection is applied. The remaining levels are based on the relevant literature [5, 25, 43] — the highest levels are approximately the same as the highest thresholds found, with the others being intervals between the null

---

[1] The window through which the user experiences the virtual environment.

setting and the highest. A rotation or translation level refers to a multiplier that is applied when real environment movement is translated into virtual environment movement. A curvature level refers to the radius of a circle in the real environment, the circumference of which must be followed to walk in a straight line in the virtual environment (∞ radius is a straight line). The order in which participants were exposed to the various techniques and gains were randomized in order to reduce bias.

Curvature Gains

The findings of previous work regarding curvature gains have been stated in terms of the radius of the circle that users walked in. For this reason, we had the idea of writing our system based on how large of a circle we wanted. The input of our curvature algorithm is therefore the radius of the circle in which we want our users to walk. However, this does not tell us how much we need to rotate the user in each frame in order to achieve this. In order to apply curvature gains to guide the user along the circumference of a circle with radius r that is centered in the middle of the real environment, it was necessary to calculate the correct rotation to add for each frame update in the virtual reality experience. We used a combination of formulas to do so.

The first formula is the time it takes to walk the circumference of a given circle. To calculate this, we assumed that the user was walking with constant velocity v. This allowed us to use the formula $t = \frac{d}{v}$ to calculate the time (t) it takes to walk a certain distance (d) at that velocity. We aimed to have the user walk the circumference of a circle, which means that $d = 2\pi r$. This gave us the following equation:

$$t = \frac{2\pi r}{v} \tag{5.1}$$

Second, we were interested in the number of frames (F) it would take to walk a full circle. If we know how much time it would take to walk the circle as well as the time between frames (ΔT), then we can find the total number of frames it would take to walk the circle by using the following equation:

$$F = t \cdot \frac{1}{\Delta T} = \frac{t}{\Delta T} \tag{5.2}$$

Third, walking the complete circumference of a circle would mean that the user walked 360 degrees of the circle. Consequently, finding the number of degrees to add per frame (θ) would mean dividing 360 degrees by the number of frames it takes to walk the circle: $\theta = \frac{360}{F}$. Substituting F from Equation 5.2 into this equation gives us: $\theta = \frac{360\Delta T}{t}$. Substituting t from Equation 5.1 into this equation gives us how much rotation must be added for each frame (see Equation 5.3). Since our program has access to these variables without further computation, the equation we are using is:

$$\theta = \frac{360V\Delta T}{2\pi r} \tag{5.3}$$

However, this formula only tells us how much we need to rotate, not in which direction. We therefore must take into consideration where the center-point of the real environment is in reference to the user. For curvature to work, the user walks in an orbit around the center, which is done by rotating the virtual environment away from the center. This must be addressed when the user tries to counteract the rotation added to the avatar, which results in a rotation towards the center of the real environment. For this purpose, we took into account the position vector of the user $\vec{U}_P$ (from the center of the room) and the user's yaw rotation $\vec{U}_R$. Note that these vectors symbolize two-dimensional space since height (position) is not taken into consideration, and that yaw is rotation in two-dimensional space. We took the cross-product between these two vectors ($\vec{U}_R \times \vec{U}_P$) to find where the center is in reference to the user (D). If the result is a positive number, then the center is to the right of the user, and conversely. We were interested only in the sign of the resulting cross-product because we only needed to know whether the resulting cross-product was positive or negative. However, we wanted to rotate the virtual environment away from the center. For this reason, we negated[2] the cross-product before finding its sign. This is indicated by Equation 5.4.

$$D = \begin{cases} 1, & \text{if } -(\vec{U}_R \times \vec{U}_P) > 0 \\ -1, & \text{if } -(\vec{U}_R \times \vec{U}_P) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

Unreal takes positive rotation as rotation to the right and negative rotation as rotation being to the left. This cross-product gives us the direction in which we would like to rotate. We therefore multiplied θ by D to obtain the final direction and rotation in which we wished to rotate (see Equation 5.5).

$$C_{\text{AddedRotation}} = \theta \cdot D \tag{5.5}$$

We employed Equation 5.5 to apply the resulting $C_{\text{AddedRotation}}$ to the user's pawn in the virtual environment. Although it is the case that users change their speed over time, our tests during development showed that, by means of this equation, the resulting curve gave the desired circle based on the input of radius r.

Rotation Gains

Since most related work in this field has reported findings regarding rotation gains as the multiplied scale of the real environment rotation, we developed our system to take such scales as an input. By multiplying the real environment rotation ($\Theta_{Physical}$) by this scale (g_r), we obtained the rotation of the virtual environment ($\Theta_{Virtual}$). This can be found by modifying Equation 3.2 from Section 3.1.2 and solving for $\Theta_{Virtual}$, which gives us Equation 5.6.

---

[2] Multiply by -1.

$$\Theta_{\text{Virtual}} = \Theta_{\text{Physical}} \cdot g_r \tag{5.6}$$

Our system uses two parameters for rotational redirection: an upper limit and a lower limit. We have emphasized finding the threshold for the upper limit in the research presented here. We therefore changed the values of the upper limit in order to find the threshold while keeping a static lower gain. The upper limit was applied when the user rotated away from the center, while the lower limit was applied when they rotated towards the center. For this reason, we needed to consider where the center-point of the real environment was in relation to the user. To do so, we took into account the position vector of the user $\vec{U_P}$ (from the center of the room) and the user's yaw rotation $\vec{U_R}$. The cross-product between these two vectors indicated in which direction the center of the room was. If the resulting cross-product was a positive number, then the center was to the right of the user, and conversely. Furthermore, Unreal takes positive rotation as being to the right and negative rotation as being to the left. Consequently, if dividing the cross-product by the change in rotation yielded a positive number, then the user was actively rotating towards the center, while a negative number meant that the user was rotating away from the center. This is illustrated by Equation 5.7.

$$\Theta_{\text{Virtual}} = \begin{cases} \Theta_{\text{Physical}} \cdot g_{r\_lower}, & \text{if } \frac{\vec{U_R} \times \vec{U_P}}{\Theta_{\text{Physical}}} > 0 \\ \Theta_{\text{Physical}} \cdot g_{r\_upper}, & \text{if } \frac{\vec{U_R} \times \vec{U_P}}{\Theta_{\text{Physical}}} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.7}$$

Translation Gain

Similar to the case with rotation gains, most related work in this field reports findings regarding translation gains as the multiplied scale of the real environment translation. We therefore developed our system to take such scales as an input.

Calculating the amount of added translation to be applied can be found by modifying Equation 3.1 from Section 3.1.1. Solving this equation for $S_{Virtual}$ gave us Equation 5.8. Multiplying the gain parameter (g_t) by the physical translation ($S_{Physical}$) should give us the virtual translation ($S_{Virtual}$) of the user.

$$S_{\text{Virtual}} = S_{\text{Physical}} \cdot g_t \tag{5.8}$$

Unlike the other reorientation techniques, translation acts independently of the center-point of the real environment. Regardless of the scenario, the applied translation is always the same.

5.2.3 Ramping

When applying rotation and curvature gains, we observed that after implementing the algorithms, solely on the basis of their mathematical descriptions, difficulties emerged. The problem we encountered was that the algorithms lacked a *smoothing* element to help transition between the gains that were applied. This phenomenon became particularly evident when the user rotated and the room's center-point switched sides relative to their orientation during rotation gain and curvature gain. In this instance, both algorithms change the direction of the applied rotation, and without any smoothing this can lead to an abrupt change that is very noticeable to the user.

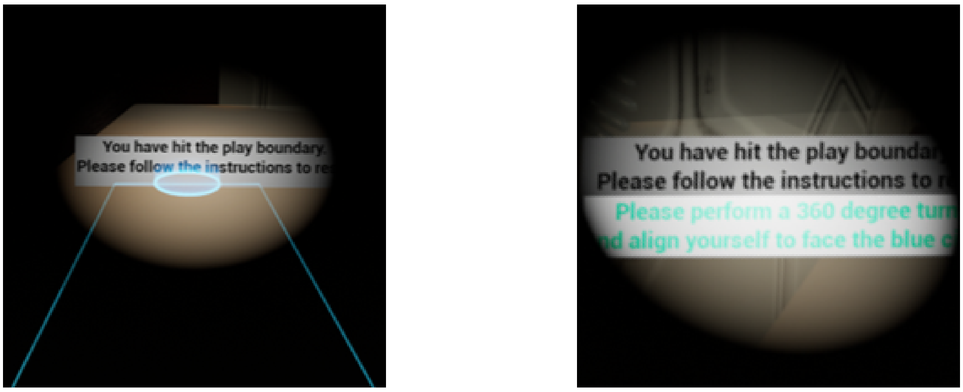
To address this issue, a ramping mechanism was implemented for both techniques. A simple step algorithm was implemented to increase or decrease the gain by a set amount in every frame for rotational gains. For instance, if we switched from a gain of 1.0 to 1.5, the ramping mechanism would smooth this transition over the course of 1 second, thus eliminating the sudden jolt of going from no gain to a larger gain.

Curvature gain depends on a defined circle with a given radius, where the neutral value is ∞, which achieves a straight line. Consequently, the ramping mechanism for curvature had to be slightly different. Specifically, when changing between rotating right or left, the gain was first ramped up to a high value (50 meter radius) before ramping down again. This gradual ramping to a neutral state in which no rotation was added, followed by ramping down to the desired radius, ensured a smooth transition. The ramp time was set to a fixed interval of 0.75 seconds.

5.2.4 Resetting

A reset was initiated when the user leaves the predefined play area. We implemented two techniques that were used when resetting the user (Figure 5.1): Look-At-Center (Figure 5.1b) and Walk-Towards-Center (Figure 5.1a). Look-At-Center is aimed at aligning the user's real environment orientation with the center of the play area. During the rotation, their avatar's rotation was scaled in such a way that, upon facing the center in the real environment, the avatar will have completed a full 360 degree turn. Look-At-Center is aimed at re-orienting the user to face the center but, does not actively move them away from the boundary. It is therefore necessary to transition into Walk-Towards-Center to move the user away from the boundary into which they collided.

During resets, the movement of the user's avatar was locked to preserve the state of the virtual reality experience when resetting. During development, we found that the effects of rotation added during Look-At-Center, with having the position of the avatar locked, could be quite disorienting. We therefore introduced a "tunnel vision" effect during resets that narrowed the user's field of view. We found that this limited some of the negative effects of resetting. In respect to curvature gains, looking directly at the center could cause the algorithm to not apply any rotation due to oscillation of the direction in which the algorithm wanted to rotate. We therefore introduced a special case in which if curvature was being applied, we aimed the user towards a point offset slightly from the center in order to steer the user onto an orbit around the center.



You have hit the play boundary.
Please follow the instructions to re

You have hit the play bounda
Please follow the instructions to r
Please perform a 360 degree tur
d align yourself to face the blue c

(a) Walk-Towards-Center

(b) Look-At-Center

Figure 5.1: What users see during resets

5.2.5 Save States

Throughout the tests, we stored the progression of the participants as a ”save state” in the event that a test was incomplete (system crash, participant needed a break, etc.). This was done by using Unreal Engine’s ”save game to slot” mechanism to store information such as the tests the participant had completed and those remaining. Whenever the participant was ready to restart the test, we loaded the save state and the participant continued from where they left off. We saved the state of only one participant at a time in order to limit memory usage. Whenever a new participant entered into the system, we deleted the old save state and created a new one for the current participant. This was done by a quick check when the program first began that compared participant identification numbers. If the two numbers were the same, we loaded the previous save state, and we created a new one if they were different.

5.3 Virtual Environments

In this study, our aim was to explore the ways in which one can convincingly perceive first-person motion in a virtual environment that is larger than the rather limited real environment in which the actual movement was happening. Ideally, the participant should be able to move around the virtual environment somewhat infinitely while not hitting any physical obstacle (objects, walls, etc.). Indeed, such virtual environments cannot be empty because the participant would then concentrate on remaining in the real environment in the absence of any stimuli in the virtual environment. We thus designed specific virtual scenes to motivate the participant to move naturally in the virtual environment. Moreover, each virtual environment was designed with a given redirected walking method in mind (see Table 5.2 for an overview). This was done in order to maximize the use of the given redirected walking technique.

Technique	Virtual Environment
Translation	Witches' Cauldron
Curvature	Track and Field
Base Rotation	Fire Drill
Distractor Rotation	Dungeon

Table 5.2: Reorientation techniques and their associated virtual environment

5.3.1 Tutorial Room

Regardless of the order in which virtual environments were presented, the first environment was always a ”tutorial world” (see Figure 5.2). We took into consideration the fact that our participants might have had little or no experience with virtual reality applications, even if they were familiar with video games. For this reason, a tutorial that helped familiarize them with basic virtual reality interaction and locomotion is essential.

Figure 5.2: The Tutorial Room virtual environment

Several factors were taken into consideration when designing this virtual “base” (the initial virtual environment that the participants entered). First of all, since users in the base environment should not be exposed to any type of reorientation, we deactivated the redirected walking algorithm in this room. This was an added measure that would allow users to experience virtual reality as it is in its current state. Consequently, we could not expand the virtual world beyond what we had in the physical world. As a safety measure, the tutorial room was designed to be smaller than the real environment in order to keep users clear of physical walls and prevent injuries.

The design of the room was based on SINLAB's laboratory at the University of Oslo in recognition of where we created our experiment. Any type of interaction needed to complete our tests can be found there. As is further discussed in the respective level designs, the user must know how to interact with buttons, grab objects, and know what to do if a ”reset” occurs (see Section 5.2.4), to name a few examples. Grabbable objects and a button were therefore placed in the room. There was also an area in the room that forced the user to ”reset” so that they would understand how that worked.

The test phase began once the users were comfortable with the basics of virtual reality. The test conductor would press a key on the keyboard to place users into the testing phase.

5.3.2 The Recap House

Participants were placed at an intermission level (see Figure 5.3) immediately after a test was completed. This virtual environment was reserved for collecting subjective responses and “resetting” the user to an optimal spot in the real environment in preparation for the next phase of the experiment. This spot was indicated by a button placed in the virtual environment.

Figure 5.3: The Intermission Level (Recap House) virtual environment

What we defined as an optimal spot was the one that gave us the most room to work with, and it was situational since it depended on the virtual environment. Some virtual environments mentioned below benefited from the user being placed at one end of the real environment, while others benefited from placing the user at the center. For example, if we knew that the user was going to walk straight at the beginning, we could place them near the edge of the real environment facing the center. To do so, we checked what the next virtual environment was in the test sequence and placed it accordingly in the recap house. Furthermore, when the user pressed the button, we checked the angle between the user’s yaw from the center of the room. This angle was used to rotate the virtual environment such that they were forced to turn towards the center so that they were able to perform their tasks.

As was the case with the tutorial room, the recap house did not use redirection. We thus designed the house to be about the same size as the playing area. To prevent users from walking outside the playing area, users were instructed in advance that they should walk towards the button, but not press it until they were told to do so. Once questioning was over, the user was told to press the button in order to proceed to the next phase of testing.

5.3.3 Track and Field

Curvature gains were often applied to prevent the participant from walking in a straight line in the real environment. In order to explore acceptable levels of curvature, we designed a test that motivated the participants to walk a long straight line in the virtual environment. We introduced a *Track and Field* arena, depicted in Figure 5.4, for this purpose. The participant entered the test at the starting line and was asked to walk towards the finish line, which was approximately 8 meters away in the virtual environment.

Figure 5.4: The Track and Field virtual environment

The test was conducted within a limited real environment (approximately 7 meter × 8 meter). The participants reached the end of the real environment at several levels of the test, and resetting was applied when they did. This procedure extended the time for the test as well as the participant's immersion experience as they became more familiar with the reorientation.

5.3.4 Fire Drill

In this test, the participant visited several stations by walking in a zigzag pattern along a corridor in the virtual environment. Rotation gain was applied only when the participants rotated their head, which allowed for a reduction of walking distances. Our test environment took into consideration the fact that the participant should perform as many turns as possible during each test phase (see Figure 5.5). For this reason, we modified the *fire drill* proposed by Razzaque, Kohn, and Whitton [32] by adding extra buttons that the participant had to press before moving to the next one. This forced the participants to make a greater number of turns, which enabled us to take greater advantage of the available real environment. In that sense, more turns yielded more opportunities to induce redirection.

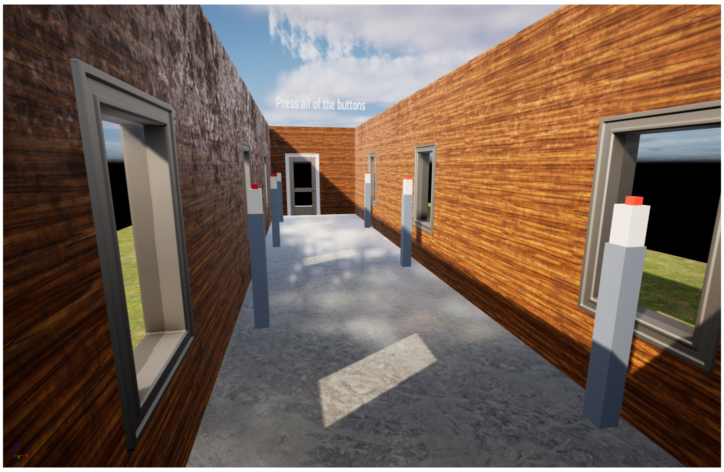


Figure 5.5: The Fire Drill virtual environment

5.3.5 Witches' Cauldron

This level was used to determine whether users were able to notice changes in the speed at which they were walking in the virtual environment. We looked for ways to reduce the number of resets needed for this level since resets might disrupt immersion for the user. In addition, since change in speed only expands the room by a factor at which the speed is scaled, how far a user could walk before being reset was limited. We therefore came up with the idea of creating a center-point, located at the center of the real environment, that the user must walk to numerous times in the virtual environment. We created a witches' hut for this purpose (see Figure 5.6).

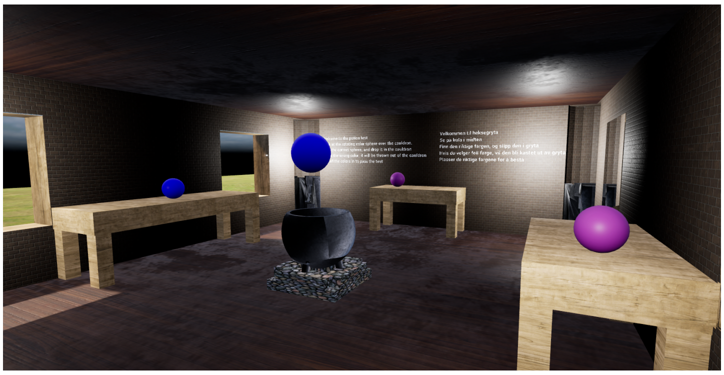


Figure 5.6: The Witches' Cauldron virtual environment

Users were asked at this level to locate the correctly colored ball, which was indicated by a ball floating above the cauldron and place it in the cauldron. A new color would then appear, and they would have to do the same. This was done four times before the test was completed. The balls were always spawned in the same location, but their color was randomized. Since each ball was subject to its own randomization, any two balls could be spawned with the same color. The order of which ball to collect was also randomized. All this allowed for the users to have multiple paths they could take. This made possible a certain degree of freedom in choosing how they would like to explore the virtual environment.

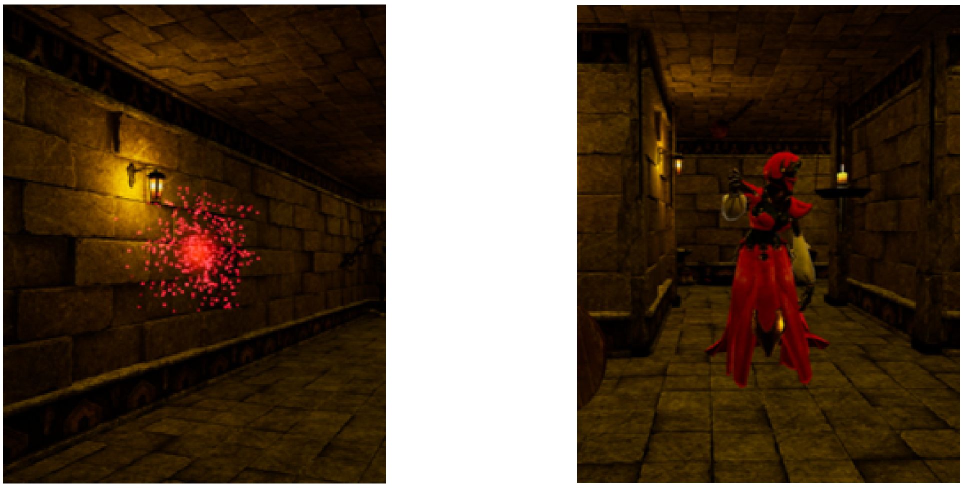
5.3.6 The Dungeon

The other virtual environments mentioned in this section were designed to be testing environments. A testing environment is a virtual environment that is designed to limit movements by the user and is aesthetically simplistic. As a result, users tend to not become fully immersed into virtual reality in such an environment. However, we were interested in observing how users experience redirected walking in a game-like environment, which is more detailed in design and allows the user to have greater freedom in walking. We created the *Dungeon* (Figure 5.7) to compare the different types of environments.

At this level, the user's task was to locate and collect the gold from three chests. These chests randomly spawned throughout the dungeon. To limit the amount of time a user took for a test, a limit of 3 minutes was added. The test was completed once all the gold was collected or the timer ran out, whichever came first.

Figure 5.7: The Dungeon virtual environment

In addition, we introduced "distractors" into the virtual environment. These distractors might or might not spawn depending on what we were testing (see Section 4.2 for details). The user would encounter two types of distractors if they spawned: (1) the dungeon keeper and (2) the wisp. The dungeon keeper (see Figure 5.8b) spawned in the virtual environment and was tasked with finding the user. Once the dungeon keeper found the user, they would continuously throw fireballs at them. In order to avoid taking damage, the user was equipped with a shield in the virtual environment that would deflect such attacks back towards the dungeon keeper. The user had to deflect two fireballs back at the enemy to defeat them. The wisp (see Figure 5.8a) spawned when the user was about to leave the playing area. It would fly across the user's face to get their attention before going into orbit around them. At any given point, the wisp would shoot a fireball at the userand then run away.



(a) The Wisp distractor

(b) The Dungeon Keeper

Figure 5.8: The various distractors used in the Dungeon

5.4 Pilot Study

We conducted a pilot study to ensure everything was working as it should before proceeding with the main tests, using a smaller real environment of approximately 4.5 meter × 4.5 meter (see Figure 5.9). We focused on the simplest tests we had at our disposal, which ended up being track and field and the fire-drill. We followed the steps described in Section 4.2, without the added step of having users register before testing. We had the opportunity to publish an article based on our preliminary results during our pilot study, and we used it to become more familiar with Microsoft Excel and how we could properly interpret the data. By doing so, we saved time for when the later studies were done since we knew how we wanted to set up our data. In addition, Excel allowed us to "update all" references in the document, which meant our analysis would already be mostly completed even when the data dynamically changed as the tests progressed.

To summarize our preliminary study, we were successful in identifying the statistical significance of rotation gain that became noticeable. The curvature gain limits, however, could not be quantified with sufficient statistical significance. This was due to large differences in noticeability, although the average noticeability dropped visibly with increased levels of reorientation. This finding was based on how redirected walking affected users' quality of experience, which showed that increased levels of reorientation had a negative impact on user's quality of experience [12].

The participants in our pilot study went through a sequence of 5 tests for each reorientation technique used, for a total of 10 tests. An entire test sequence took about 20 minutes to complete. Participants were permitted to provide feedback at any point during the test regarding what they were experiencing, which we noted down. Most of their feedback concerned minor connection issues and the number of resets needed to complete the

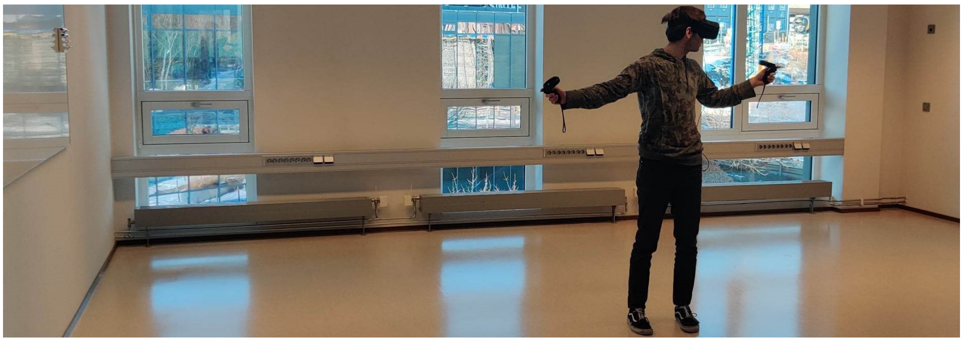


Figure 5.9: The real environment used for pilot testing

fire drill. We also observed that participants completed the track and field arena much quicker than anticipated and that there were numerous fatal GPU crashes. Post testing, we noted that very few participants noticed some of our gain parameters, and that our questionnaire apparently reported fewer data points than we had intended. We therefore made the following changes to our study design:

GPU crashes GPU crashes occurred when the computer required a lot of resources. Under-clocking the GPU[3] seemed to have narrowed it down to an acceptable level.

Connection issues We initially had a short Ethernet cable that connected the computer to the router when running our tests. Using a longer Ethernet cable allowed us to place the router more optimally and have better connections.

Track and Field edits The finish-line was pushed back about twice the distance of its original placement to give participants more time to notice the redirection.

Fire Drill edits The distance to walk between the buttons was originally quite long. We reduced it to limit number of resets.

Intermission edits Initially, the *progress button* was always in the center of the virtual intermission room. We decided, however, to make this button spawn dynamically within the virtual environment in order to optimize the available area in the real environment. We hoped that this would (a) give us more room for free movement in the real environment and (b) reduce the number of resets.

Questionnaire changes We had not been able to note everything that we felt was necessary in our questionnaire. We therefore changed it somewhat, such as by adding a notes section for each question, to obtain more data points.

---

[3] Reducing the performance of the GPU.

Change of parameters The original parameters used in our study appeared to have a low detection rate during the pilot. We therefore increased several of them.

Calculating Curvature Gain We observed that the desired radius that we used as input yielded a walked circle of wrong size. This resulted from a mathematical error and was fixed quite quickly.

Chapter 6

Analysis of tests

This chapter is dedicated to analyzing the various tests we have conducted for this thesis. We used a real environment of approximately 7 meter × 8 meter for our tests (Figure 6.1), and the participants were asked to complete various tasks in a virtual environment that exceeded these boundaries. As a safety measure, a buffer zone of approximately 1.5 meter was added between the play area and the physical wall in case the user moved too quickly through the boundaries of the play area.

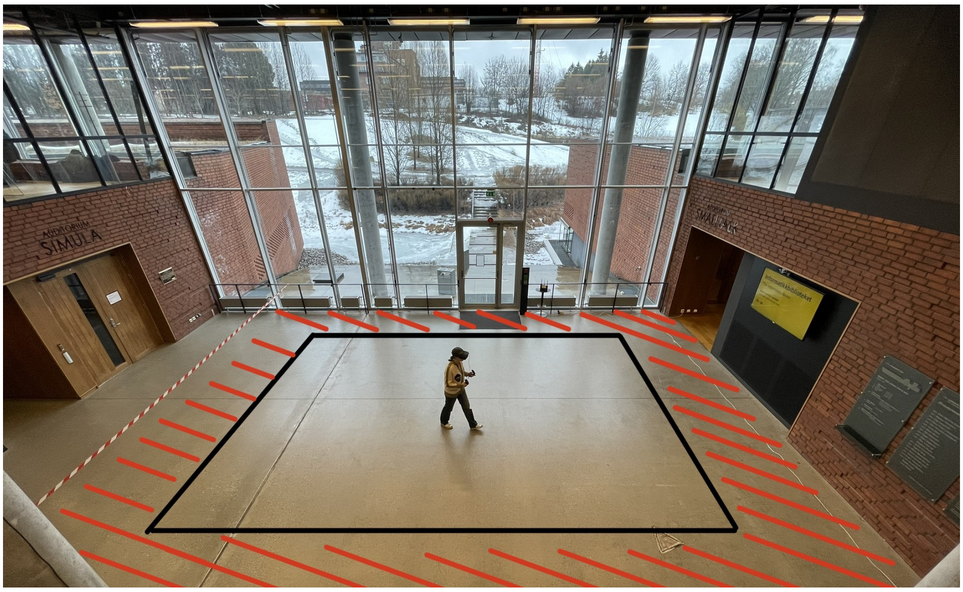


Figure 6.1: The real environment used in this research

This thesis has merged two smaller user studies and analyzed them both separately. In addition, a third analysis has been conducted which used data from the two user studies. As noted in Section 4.2, these two studies consist of the base test and the distractor test. In addition, we analyzed the differences in virtual environments between the two tests. We therefore structure this chapter into three sections, one for each type of analysis we have performed. Each section is further divided into subsections in which we analyze each factor separately, followed by a discussion.

- Analysis of Base Test (Section 6.1)
- Analysis of Distractors Test (Section 6.2)
- Analysis of Virtual Environments (Section 6.3)

The aim of this chapter is to gain a better understanding of how much we can use to manipulate visual stimuli before the users find that it negatively impacts virtual reality.

6.1 Analysis of Base Test

The sample size of our base study consisted of 16 participants, one of whom had to leave the study before we could to collect a sufficient amount of data from them. We thus did not use this participant in our analysis, leaving us with 15. In addition, we had a computer crash during one of the tests. Although we redid the test, an error on our side did not return one participant to where they had left off. As a result, they did not complete all the curvature tests, but did complete the other tests. We consequently disregarded this participant only in our analysis of curvature gains.

	df_1	df_2	F-value	F-tab	p-value
Rotation Q1	4	70	2.77	2.50	< 0.05
Rotation Q2	4	70	2.46	2.03	< 0.10
Curvature Q1	4	65	3.25	2.51	< 0.05
Curvature Q2	4	65	2.93	2.51	< 0.05
Translation Q1	4	70	0.52	0.52	0.72129
Translation Q2	4	70	1.18	1.18	0.32717

Table 6.1: Base test — ANOVA results for the different conditions

Our 15 participants had an average level of video game experience, but did not have previous experience using walking as a form of movement in virtual reality. Each of the 15 participants in our pilot study performed a series of 15 tests, 5 tests for each of the three types of reorientation. We used ANOVA to determine whether there were statistically significant differences between responses given on the questionnaire (see Table 4.2). We analyzed the results in respect to each of the three redirection techniques, which created nine ANOVA tables (see Table 6.1). A few participants showed signs of sickness in one of the tests we performed. Because of the small number of such incidents, we removed sickness (*Q3*) as a factor in studying the quality of experience.

6.1.1 Translation Gain

Figure 6.2 shows that participants were unlikely to notice translation levels ranging from 1.0 (no change applied) to 1.25 (walking 25% faster in the virtual environment). This is shown by the low mean for noticeability and the low standard deviation (black whiskers) over the various gains. ANOVA also reflects this (Table 6.1), and we cannot confirm any statistically significant differences regarding detection thresholds over the various gains. In all cases, the mean noticeability (including the standard deviation) remains below 50%. For the highest applied gain, only 2 of the 16 participants (13%) noticed a difference in their movements between the real and virtual environments. Because of the low number of such occurrences, the data suggests that users did not notice the applied redirection while walking 25% quicker in the virtual environment.

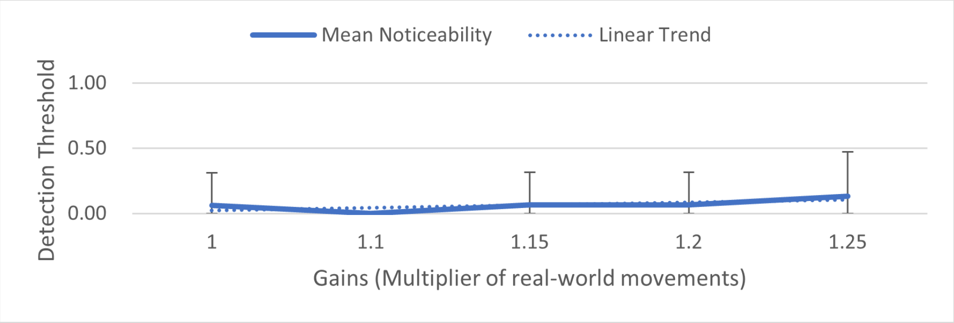


Figure 6.2: Translation Gain — Mean reported detection rate

The low detection rate regarding Q1 has promising results for translation gain. This is due to the fact that a low detection rate often yields a high quality of experience. This can be seen in Figure 6.3, where the mean quality of experience is high for all translation levels. In addition, the standard deviation over each level is quite small. The standard deviation for the two higher levels is larger than the others due to outliers rating their experience as quite poor. Even when their reported experience was taken into consideration, however, the standard deviation at each level suggests that all levels yielded a high quality of experience. For example, 93% of participants reported an experience level ≥ 4 for the highest gain. As suggested by the small standard deviation between gains, the resulting ANOVA (Table 6.1) indicates that the differences over the various gains are not statistically significant.

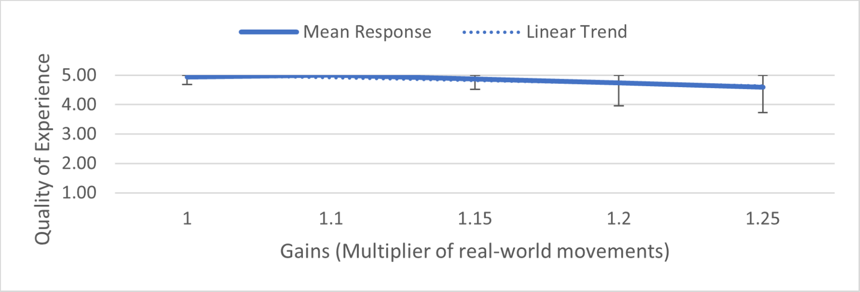


Figure 6.3: Translation Gain — Mean reported quality of experience

In the light of the above considerations, the data suggests that increasing the translational speed in the virtual environment by 25% did not lower the quality of experience in virtual reality and preserved the immersion. Furthermore, the low detection threshold in our data suggests that it should be possible in theory to increase the translation level before users notice it and find it unpleasant. If this is the case, then it should be possible to find new detection thresholds for this gain above 1.25.

6.1.2 Curvature Gain

The analysis of the detection threshold began with asking the participants whether they noticed any type of manipulation of their movements between the real environment and the virtual environment. We looked at the detection rate, that is, the number of participants reporting unnatural movements, for this purpose (see Figure 6.4). The data in this figure indicates that the linear trend of detection increased as the level of gain increased. The same holds true regarding standard deviation (black whiskers), which increased significantly for the higher gains. The large difference in standard deviation regarding gains is statistically significant with a 95% confidence (see Table 6.1). In addition, a large number of the participants noticed the curvature level of a 5 meter radius (50%), with the other curvature levels displaying a detection rate of approximately 33% or less. The standard deviation for the 7.5 meter radius is too large to provide a valid option at this point. A conservative estimate would therefore be that it is possible to walk on the circumference of a circle with a 10 meter radius due to the low detection rate and standard deviation. However, it might be possible to reduce the radius depending on the participants' quality of experience with the higher gains.

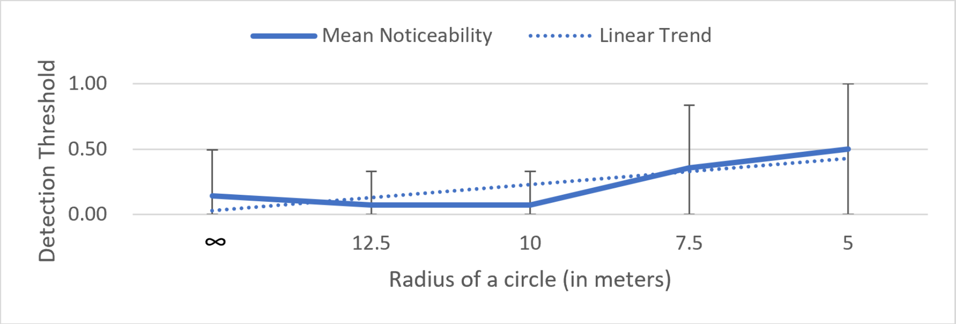


Figure 6.4: Curvature Gain — Mean reported detection rate

Analysis of the detection threshold does not conclude with whether or not participants noticed an applied gain. Our study also addressed whether a participant found an applied gain to be enjoyable, even if it had a high detection rate. Similarly to how people can change sensitivity settings, field-of-view, and many other adjustments in a video game, we believe that immersion can still be maintained if users detect the gain as long as it is enjoyable. We therefore analyzed the degree to which participants enjoyed the different levels of gain. Figure 6.5 depicts the participants' quality of experience in virtual reality for different levels of curvature gain, the graph indicating a downward trend in their quality of experience as the levels of gain increase. It is interesting that the standard deviation over the lower gains was of a lesser degree than the higher gains, who have a large standard deviation. This change in standard deviation indicates a statistical significance, which ANOVA further confirms with 95% confidence (Table 6.1). Our conservative estimate that users can walk a circle with a 10 meter radius holds true since it has the highest reported quality of experience in addition to the lowest standard deviation. A closer look at the data reveals that the large standard deviation seen with a 7.5 meter radius was caused by a single participant rating their experience with the lowest possible value (1). Removing this participant from the data set reduces the standard deviation on this gain to a value within the acceptable range (≥ 4), as Figure A.1 illustrates. In spite of this outlier, the data report that 12 of the participants 14 (86%) rated their experience as acceptable for this gain. Because of the high reported number of participants who rated their experience as acceptable — and with a low standard deviation if the outlier is removed — the data suggests that it should be possible for users to walk the circumference of a circle with a 7.5 meter radius.

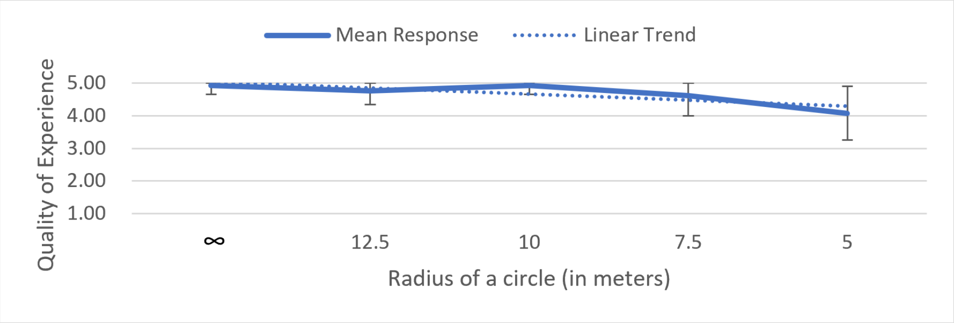


Figure 6.5: Curvature Gain — Mean reported quality of experience

Given the above considerations, we may say that a conservative estimate regarding this gain would be that users can walk the circumference of a circle with a 10 meter radius. This radius preserves immersion by neither being detected, nor lowering the quality of experience with virtual reality. Another suggestion can also be made, however. As noted above, using curvature gain with a 7.5 meter radius did not negatively impact a user's experience in virtual reality, with the exception of one outlier, although this gain became noticeable to the user if the standard deviation was considered. In this thesis, we emphasize what users find to be acceptable over what they notice. For this reason, the data tell us that we can redirect users by the means of curvature gain with a circle as small as one with a 7.5 meter radius.

6.1.3 Rotation Gain

As Figure 6.6 indicates, the reported detection rate follows a linear growth as the gains increase. Furthermore, we see that the standard deviation (black whiskers) became quite high for the higher gains (1.4 and 1.55). This reflects the varying response as the mean noticeability for these gains are close to 50%. The remaining gains (1, 1.1 and 1.25) have a reported detection rate of approximately 25% or less with a smaller standard deviation. These differences are statistically significant with 95% confidence (Table 6.1). The high standard deviation and reported noticeability suggest that the upper two gains are not valid options at this point. The remaining gains (outside of the null setting) have a high standard deviation that crosses the 50% mark, which makes it seem unlikely that either option is valid. Both gains are similar to each other in noticeability, but slightly different in standard deviation. A conservative interpretation of the data would suggest that a gain of 1.1 is the better option, although a gain of 1.25 might also work.

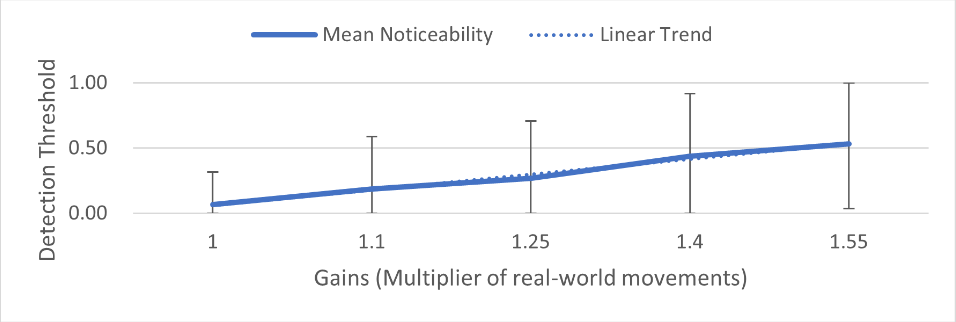


Figure 6.6: Rotation Gain — Mean reported detection rate

Figure 6.7 presents participant quality of experience in virtual reality over the range of gains for this method of redirected walking. In spite of the high reported detection rate noted above, the higher rotation gains still have a mean reported quality of experience close to or higher than what we deemed to be acceptable (4). This reflects the fact that even when participants noticed an applied gain, they still believed that it did not negatively affect the overall virtual reality experience. While it is still difficult to suggest that these gains are valid options due to the large standard deviation, the smaller gains have a noticeably better standard deviation that it falls within the acceptable range (≥ 4) in nearly all cases. Table 6.1 illustrates that these differences among gains are statistically significant with 90% confidence. A closer look at the data reveals that 13 of 15 participants (87%) rated their experience as acceptable with a gain of 1.25. Because of this high value, the data suggests that we can use a gain of 1.25 for this method of redirected walking. There is an outlier in our data set, however — one of our participants reported multiple low scores while others rated it more highly. For example, although 12 of 15 participants (80%) reported that a gain of 1.4 was acceptable, this participant ranked their quality of experience with the lowest possible value (1). The standard deviation for this gain changes significantly when we remove this participant from our population — Figure A.2 shows that the standard deviation then falls close to an acceptable value. Furthermore, the percentage of participants rating this gain as acceptable increases to 86% when we eliminate this participant. With this in mind, an aggressive interpretation of the data suggests that we can use a rotation gain of 1.4.

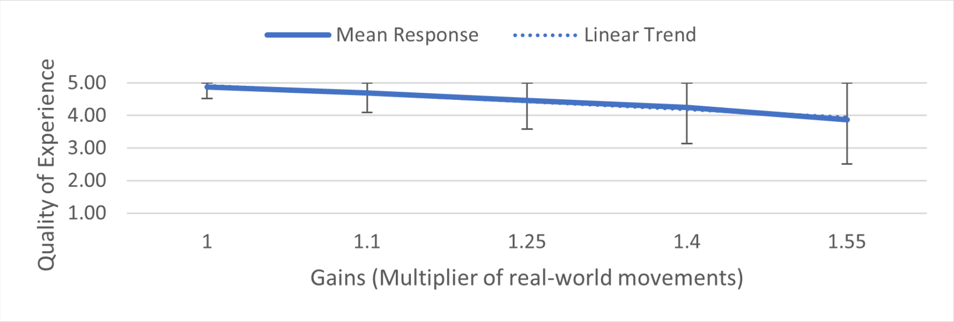


Figure 6.7: Rotation Gain — Mean reported quality of experience

In light of the above considerations, the results suggest that users can be rotated 25% more in the virtual environment than in the real environment, which is a gain of 1.25. While this gain was somewhat noticeable, it did not ruin the experience of virtual reality. We can also make another suggestion. If we disregard the outlier, then users can be rotated as much as 40% more in the virtual environment than in the real environment. Although this has a high detection rate (close to 50%), it did not impact the virtual reality experience. Since we emphasize game experience over detection in our thesis, our results then suggest that we can rotate the user with a gain of 1.4.

6.1.4 Discussion

For all three redirected walking methods, we observed trends which show that the participants' virtual reality quality of experience was negatively correlated to the gain levels. This confirms our expectation that the level of transparency of redirected walking declines as the divergence between movement in the real environment and the virtual environment increases. In addition, both methods yielded an increasing standard deviation that is proportional to the increase in the levels of gain. Further analysis is necessary if we wish to distinguish and generalize the contribution of situational differences in respect to human tolerance.

Our goal in this thesis was to push the limit regarding how much we could manipulate users' senses before this negatively impacted their experience in virtual reality. For this purpose, we examined prior work to find reported thresholds that we could use in our tests (see Section 3.1).

Rotation Gain Steinicke et al. [43] suggest that participants can rotate approximately 49% more in the virtual environment than in the real environment. Since our findings suggest that we can rotate to only

40%, we cannot confirm Steinicke et al. [43] claim. This is in part due to our linear approach to selecting various gains by increasing 0.15 for each gain. We consequently did not test the gain of 1.5, but rather 1.55, which failed. If Steinicke et al. [43] assessment is true, then the detection threshold should be somewhere between 49% and 55%.

Curvature Gain Grechkin et al. [5] suggest that participants can walk the circumference of a circle with a radius of as much as 11.6 meters or as low as 6.4 meters without noticing it. Our findings suggest that it is possible to walk the circumference of a circle with a 7.5 meter radius. We therefore cannot confirm Grechkin et al. [5] smaller 6.4 meter radius. We were, however, successful in reducing their larger 11.6 meter radius. We should note that our participants walked at their "normal" walking speed. Kloos et al. [13] report that the speed at which a user walks impacts the radius of the arc since users are less sensitive to noticing curvature gains at slower speeds. Variations in results may then be due to methodological differences in the research such as users' walking speed [5].

Translation Gain Steinicke et al. [43] suggest that we can speed up user movement in the virtual environment by 26%. Our findings suggest that we can do so by 25%. Furthermore, we suggest that this can be increased even further due to the low reported detection rate and high quality of experience found in our research. We therefore confirm Steinicke et al. [43] suggestion, but believe this can be increased further.

6.2 Analysis of Distractors Test

The sample size of our distractor study comprised 16 participants. These participants had an average level of video game experience, but did not have any previous experience using walking as a form of movement in virtual reality. Each of the 16 participants in our distractor study performed a series of 8 tests (4 tests with distractors and 4 without).

	df_1	df_2	F-value	F-tab	p-value
No Distractor Q1	3	60	4.00	2.76	< 0.05
No Distractor Q2	3	60	1.98	1.98	0.10701
With Distractor Q1	3	60	1.60	1.60	0.18398
With Distractor Q2	3	60	2.51	2.18	< 0.10

Table 6.2: Distractor test — ANOVA results for the different conditions

We used ANOVA to determine whether there were statistically significant differences between responses on the questionnaire (see Table 4.2). We analyzed the results in respect to each of the two conditions, which yielded six ANOVA tables (see Table 6.2). Since only a few participants showed signs of sickness in one of the tests, we removed sickness (*Q3*) as a factor for studying the quality of experience in light of the low number of such incidents. In order to see how effective distractors were in redirected walking, we had to compare results to a sanitized (no distractors) version of the scenario. We therefore needed to analyze how effective rotation gains were with and without distractors and then compare the results in the discussion.

6.2.1 Rotation Gain without Distractors

Figure 6.8 indicates that participants are more likely to notice the reorientation that has been applied when the levels of rotation increase. The reported mean of noticeability followed a linear trend such that the standard deviation (black whiskers) increased as the level of gain increased. Table 6.2 shows that these differences are statistically significant with 95% confidence. The mean reported for noticeability remained below 50% in all cases. With the added standard deviation, however, the highest gain (1.5) nearly reached 100%. It is therefore risky to claim that such a level of gain is usable without looking at the users' quality of experience. A more conservative estimate would be to use a gain of 1.25 since it is ≈ 50% with the added standard deviation.

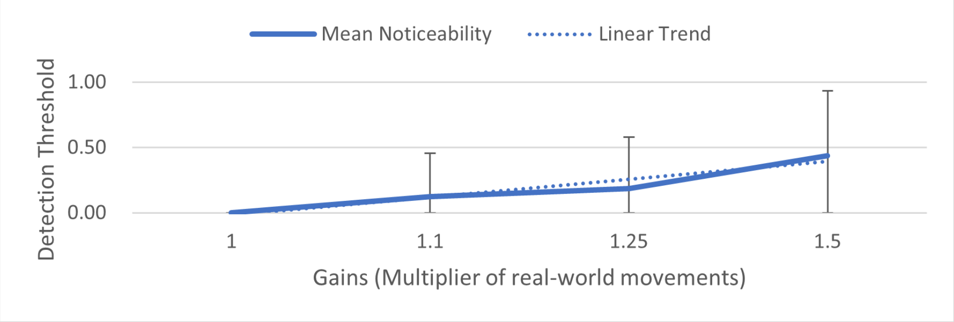


Figure 6.8: Rotation Gain without distractors — Mean reported detection rate

Figure 6.9 depicts the participants' quality of experience in virtual reality over different levels of gain. We found that there was a downward trend regarding participant quality of experience as the levels of gain increased. Similarly, the standard deviation (black whiskers) increased as the levels of gain increased, although the difference in the standard deviation was quite small. This is reflected in the ANOVA presented in Table 6.2, and we could not confirm any statistically significant differences in user experience between the levels of gain. As for our estimate regarding Q1, the quality of experience with a gain of 1.25 fell within the "acceptable" range, with only a small portion of the standard deviation falling below the threshold (≥ 4). The same was also true for the highest tested gain of 1.5, where 88% of participants reported a quality of experience with a level ≥ 4. Because of the high number of acceptances concerning quality, our results suggest that users enjoyed their experiences in virtual reality when exposed to a rotation gain of 1.5.

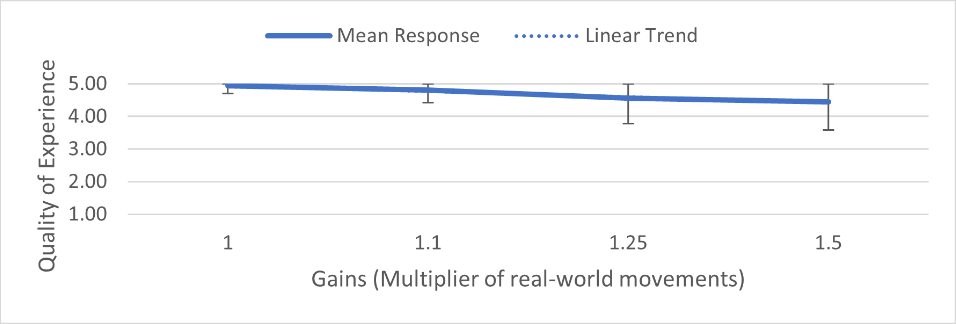


Figure 6.9: Rotation gain without distractors — Mean reported quality of experience

Most related work reports findings on the basis of whether users noticed the injected redirection. However, most applications that utilize movements make it possible for users to change their sensitivity levels to permit personalization. We therefore believe that redirected walking should place greater emphasis on the level to which it is enjoyable (how usable it is). As a result, high detection thresholds can be overlooked if users still believe that it is enjoyable to use it. This is the case we see here, when the detection threshold sat at $\approx$ 50% but users still found it enjoyable to use. We therefore suggest, based on the data presented, that we can rotate the user 50% more in the virtual environment without ruining the user's experience in virtual reality.

6.2.2 Rotation Gain with Distractors

Figure 6.10 displays how often participants noticed applied redirection. It is interesting that 25% of participants reported that their movements in the real environment did not correlate with their movements in the virtual environment when no redirection was applied. This is higher than for the two lower rotational levels tested in this study, namely, 10% and 25% added rotation. Nonetheless, the linear trend increased as the gains increased, indicating that users noticed the redirection more as the gains became larger. Furthermore, the standard deviation (black whiskers) indicates that responses varied more for higher levels (ignoring the null setting). Because of the high standard deviation for the highest gain (1.5), it is risky to suggest that such a gain can be applied. A lower gain, such as 1.1, is a safe option since the mean noticeability is close to 0, with the standard deviation being approximately 0.25. The middle option of a 1.25 gain can be realistic since the mean noticeability was less than 0.25 and only a small portion of the standard deviation exceeded the 0.5 threshold. As seen by the high standard deviation within the levels, we cannot confirm any statistically significant differences between levels for this analysis with ANOVA (see Table 6.2).

Figure 6.11 depicts the participants' quality of experience in virtual reality over different levels of gain. We noticed that there was a downward trend regarding participant quality of experience as the levels of gain

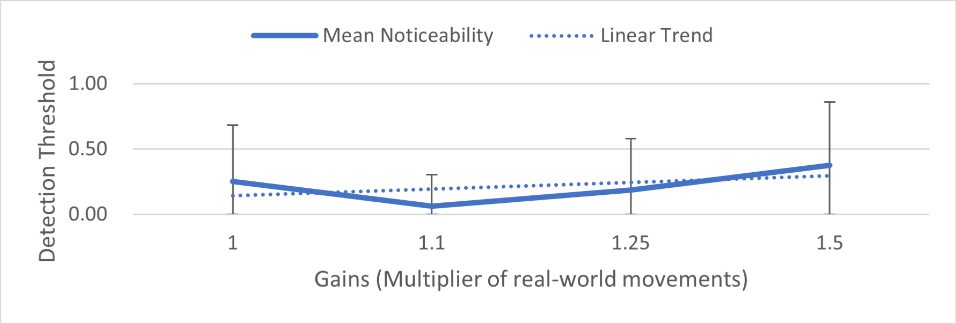


Figure 6.10: Rotation gain with distractors — Mean reported detection rate

increased. The standard deviation (black whiskers) also increased as the levels of gain increased, and there was a much larger standard deviation at the highest gain (1.5) than at the other levels. These differences are statistically significant with a 90% confidence (see Table 6.2). Concerning our estimate from Q1, the quality of experience with a gain of 1.25 fell within the ”acceptable” range, with only a small portion of the standard deviation falling below the threshold (≥ 4). This is unlike the case for our highest gain of 1.5. While the mean response fell within the acceptable range, the large standard deviation fell as low as ≈ 3. In addition, 75% of participants reported a quality of experience level ≥ 4. With this in mind, it is unrealistic to suggest that applying such a gain will not affect the virtual reality experience. A conservative approach suggests that users will enjoy their experience in virtual reality while being exposed to a rotation gain of 1.25.

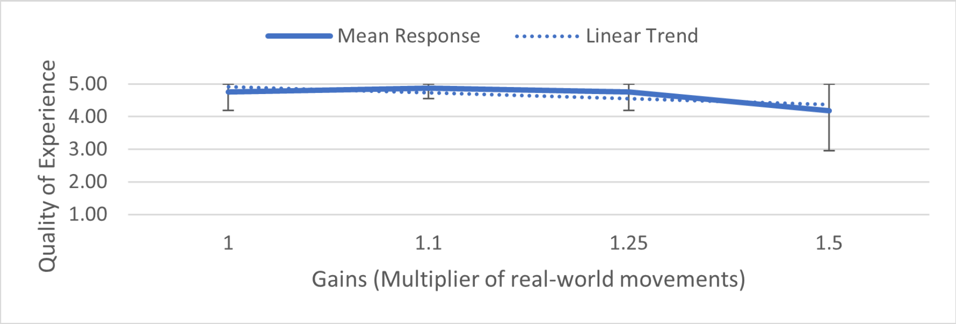


Figure 6.11: Rotation gain with distractors — Mean reported quality of experience

As a baseline, we placed users into our testing environments without any applied redirected walking. It is interesting that 25% of participants reported that their movements in the real environment did not correlate with what was happening in the virtual environment. In spite of this, we noticed a high quality of experience for this gain. This reveals the importance of distinguishing detection from quality of experience as well as why we analyzed the data from this point of view. With this in mind, it is too risky to suggest that it is acceptable to use the highest gain (1.5) because of the reported quality of experience at that level. We therefore suggest, based on the data presented, that we can rotate the user 25% more in the virtual environment without ruining their experience in virtual reality.

6.2.3 Discussion

Our hypothesis in respect to the use of distractors was that we could rotate the user more in the virtual environment (or see better performance) when distractors were involved. To test this, our analysis of the use of distractors focused on whether (a) we could use higher gains when redirecting with distractors or (b) see better performance over the different gains when distractors were involved. In their own respective subsections, we provided an assessment of the highest usable gain for both with and without distractors in the virtual environment. In Section 6.2.1 we concluded that it should be possible to rotate the user 50% more in the virtual environment and still have it be enjoyable to use when there were no distractors. However, in Section 6.2.2 we concluded that it should be possible to rotate the user 25% more in the virtual environment and have it remain acceptable to use when distractors were involved. Based on these assessments, we may conclude that there is a difference between the two conditions. These findings do not support our initial hypothesis and we wish to take a closer look at why this is the case.

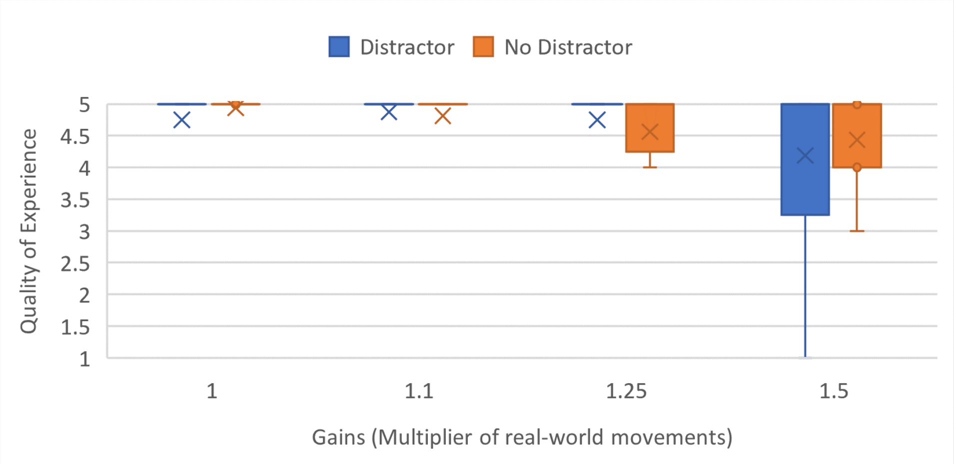


Figure 6.12: Distractors comparison — Mean reported quality of experience

Figure 6.12 illustrates user response concerning acceptability over the different scales for each scenario. As shown in this figure, our hypothesis holds true for the lower gains (between 1 and 1.25), but fails for the higher gain (1.5). The two scenarios — with and without distractors — fall nowhere close to being the same. We believe that the issue here concerns what we term the *high gain fallacy*. Research on redirected walking has often tried to find the highest possible threshold that does not break immersion. However, how usable such thresholds are in a real scenario, such as gaming, is often overlooked. High gains are desired when moving long distances, but fail when accuracy is needed — as eluded to by Razzaque, Kohn, and Whitton [32]. Applications such as virtual house tours, in which users would use redirected walking, require the ability to pinpoint movements, hence the fallacy. In our test, it may be the case that when the user attempted to follow the wisp, for example, they lacked the pinpoint accuracy needed to do so. By this we mean that they tried to follow the wisp's movements, but rotated too far because a high gain became noticeable to them.

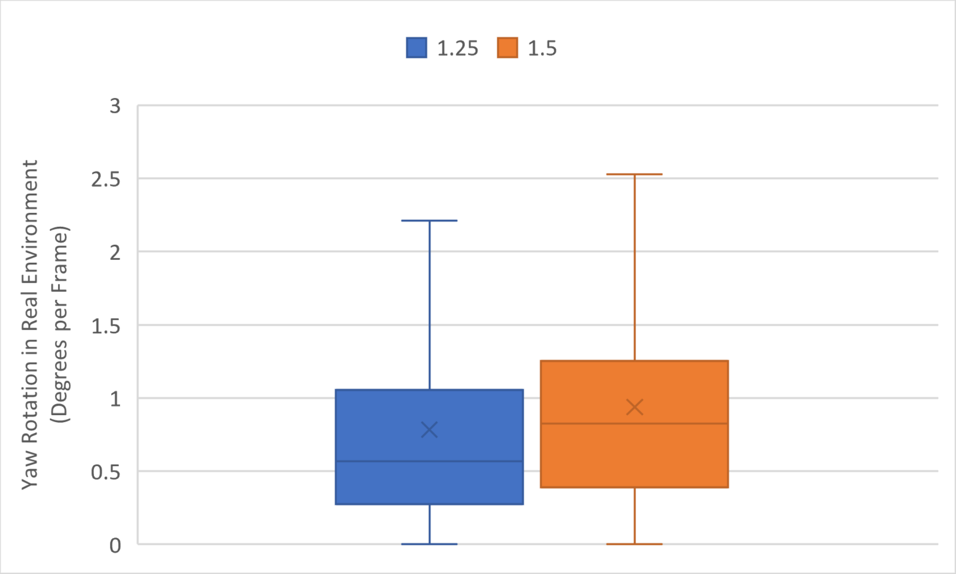


Figure 6.13: Distractors comparison — Change in rotation

If our "high gain fallacy" theory is true, then there should be a discrepancy regarding user movements at different levels of gain. Figure 6.13 illustrates a single participant's change in rotation in the real environment over the duration of a wisp with a rotation gain of 1.25 (blue) and 1.5 (orange). It should be noted that the speed of the wisp did not change between the different levels of gain. If our theory is true, we would expect higher rotation and a less dense box plot for the higher gain, which would indicate jitters (struggling to stay on course). For the lower gain, we would expect a denser graph, with data points closer to each other, since the user was able to "flow with the wisp." This would normally be the opposite, however, since a higher gain would mean that less rotation in the real environment was needed to keep up with the rotation of the wisp.

Figure 6.13 illustrates that the gain of the 1.5 box[1] is higher than the lower gain of the 1.25 box. In addition, the whiskers[2] for the higher gain are higher than for the lower gain. Keeping both of these facts in mind reveals that the user rotated more when following a wisp when higher gains were applied than for lower gains. The speed of the wisp remained the same,

---

[1] The *box* in a box plot indicate the data that is between the lower quartile (25% of scores) and upper quartile (75% of scores).

[2] The *whiskers* in a box plot indicate the lowest and highest score, excluding potential outliers.

however, as noted above. On the basis of this assessment, it appears to be the case that users were more sensitive in respect to fine-tuned movements. Our analysis thus indicates that these types of movements interfered with the experience of virtual reality since users were able to detect differences in their movements between the two environments when precision was needed.

6.3 Analysis of Virtual Environments

This section addresses how different virtual environments affected redirected walking. Both of the two virtual environments that we compared, the dungeon virtual environment and the fire drill virtual environment, used rotation gains as their redirected walking method. An analysis of rotation gain for the fire drill (Section 6.1.3) and the dungeon (without any distractors) (Section 6.2.1) are provided in their own respective sections. In summary, we found that we could rotate the user 40% more in the fire drill virtual environment, but 50% more in the dungeon. This alone tells us that redirected walking (with rotation gain) is affected by its virtual environment. We believe, however, that this deserves further analysis. Of the 22 participants in our research, 9 participated in both of our studies. These 9 comprise our sample size for taking a closer look.

	df	χ^2	p-value
Q1	5	0.0305	0.999
Q2	5	0.0426	0.999

Table 6.3: Comparing virtual environments — Chi Squared results (χ^2)

Table 5.1 indicates that levels of gains are different between the two tests we performed. Although we cannot directly compare the two tests due to the difference in the higher levels of gain, the lower gains (1, 1.1, and 1.25) are the same for the two tests. We therefore first checked to see whether the ″sample″ represented the ″population″ in order to properly assess the effects of the higher gains. In order to do so, we needed to ″manipulate″ the data so that our two data sets (sample and population) are of equal length. As seen in Appendix C, we achieved this by grouping the responses in accordance with the virtual environment and the respective gain. The responses were then averaged out to obtain a value for that group. Running a χ^2 test on these data sets gave the results displayed in Table 6.3, which show that we cannot reject our null hypothesis for either question. Consequently, we cannot say that differences between sample and population are statistically significant. On this basis, we could assume that our sample size represented the population. Since this was the case, we could use the observed responses of the population to compare the different virtual environments for the higher gains. For this reason, our initial assessment that differences in virtual environments affect the use of redirected walking holds up. However, it was not fair to use this comparison because the levels were different for the higher gains. We should therefore make an assessment to see whether users would have noticed a rotation gain of 1.5 during the fire drill. If they would not have noticed this level of gain, then we could conclude that redirected walking is not affected by its virtual environment.

As discussed in Section 6.1.3, the data did not suggest that we could use a gain of 1.4 when analyzing the detection threshold. It was likely, due to the linear trend, that a higher gain would be noticeable to the user since this gain was noticeable. This is further confirmed by the increasing linear trend in Figure 6.6 which shows that users are more likely to notice higher gains. Furthermore, the linear trend shown in Figure A.2 indicates that higher gain results in lower quality of experience. With this in mind, a gain of 1.4 already had a standard deviation (black whiskers) just under what we deemed to be acceptable. If we accept the trend to be true, we may expect that the mean response decreased, and the standard deviation increased for a gain of 1.5.

Figure A.2 indicates that the trend line intercepts the mean response line approximately halfway between the points 1.4 and 1.55. The trend thus suggests that a gain of 1.475 should be the average between the two points. If so, the data predict that the mean response for a gain of 1.475 would be 4.36, with a standard deviation of 0.718. This would place the lower whisker at a quality of experience of 3.64. As a comparison, a rotation gain of 1.5 in the dungeon had a mean quality of experience of 4.44 and a lower whisker of 3.58. These numbers show that the mean response was better in the dungeon, while the standard deviation was better for the fire drill. However, this predicted gain from fire drill was still lower than the 1.5 at which we are comparing it to the dungeon. We therefore need to take into consideration that the mean response will drop slightly and the standard deviation will increase somewhat as we approach a gain of 1.5. For this reason, the standard deviation in both cases can be predicted to be close to the same, while the mean quality of experience would be lower in the fire drill virtual environment.

On the basis of the analysis provided above, the data suggests that redirected walking, specifically rotation gain, is affected by how immersed the user is in the virtual environment. Our predicted outcome for fire drill comes close to the same quality of experience we see in the dungeon, but falls short in respect to the mean response.

Chapter 7

Conclusions

7.1 Thesis Summary and Conclusion

Virtual reality is advertised as an escape from reality and the exploration of new worlds. Movements in the real environment are mimicked in a virtual environment in a way that creates the sensation that you are in that virtual environment. This is the essence of virtual reality — making a person believe that the virtual environment that they are in is reality. The first steps towards creating a virtual reality can be traced back over 2000 years ago, when philosophers such as Plato and his student Aristotle debated on how something can be determined to be real. In this thesis, we follow what Aristotle defined as real: the way in which we process the pieces of information that we obtain from our senses is what determines whether or not something is real [6]. Aristotle described five of the senses that provide us with information: sight, hearing, touch, taste, and smell. Fast forwarding 2000 years later, we see researchers exploring the ways in which we can use these senses to alter reality, including virtual reality.

There are numerous ways users can move through virtual reality. This includes, but is not limited to, using joysticks or omni-directional treadmills and walking. The issues with most forms of movement are its high cost and whether it feels unnatural [32]. The most natural form of movement is coincidentally also the least expensive, namely, walking. Most, if not all modern day head-mounted displays — or virtual reality goggles — include tracking of where one is in a room. This allows users to walk freely in the real environment in order to move in the virtual environment as mapped 1:1. However, limitations upon available space in the real environment affect how much of the virtual environment a user can explore. Ideally, we would like to move infinitely in the virtual world without worrying about the confines of the real environment.

Redirected walking is a research topic that investigates how we can expand the finite space available to us in the real environment in order to explore a much larger space in the virtual environment. We achieve this by decoupling the traditional 1:1 mapping between the real environment and the virtual environment, as is the case in most modern-day virtual reality applications. However, humans can to a certain degree notice changes in the mapping if it is not regarded as natural. More natural movements may be identified by manipulating human senses and observing whether movements in virtual environments are viewed as true, even though a different set of movements are being performed in the real environment.

Different forms of reorientation have been discovered, but the conditions that impact a user's detection thresholds are still not fully known. In this thesis, we explored the various ways in which we can reorient the user and the extent to which we can perform such reorientation before it negatively impacts the virtual reality experience in different conditions. We have addressed the basics of the human senses, the various redirected walking techniques, how we can force redirection through distractors, and the process that took place in creating our research.

Our research has been successful in proving that redirection is possible, and that a practical approximation of user motion is thus possible. We have also been successful in identifying a statistical significance for when curvature gain becomes noticeable. Translation gain limits cannot be quantified with sufficient statistical significance, however, even though we are able to identify translation levels that go undetected by the user. In addition, we investigated the detection threshold for rotation gain, and how the threshold might change if distractors are present in the virtual environment. Moreover, our study has successfully identified the statistical significance of rotation gain that becomes noticeable under these conditions. Finally, we used linear regression to evaluate how this threshold may change depending on the virtual environment. We concluded that users are less likely to notice redirection when they are invested in the virtual environment. This finding is based in this thesis on how redirected walking affects a user's quality of experience, which indicates that increased levels of reorientation have a negative impact on users' quality of experience.

7.2 Future Research Orientations

There are a number of ways in which the research presented in this thesis could be extended. These include the following issues:

Manipulation of other Senses Out of the five senses described by Aristotle, only two have been researched in respect to virtual reality. Since reality may be defined by how we perceive differing sensory information, more research should be conducted on finding the effectiveness of manipulating the other three senses.

Combining Algorithms Most current research concerning redirected walking emphasizes the use of one redirection method at a time. It is not yet clear how it may be possible to combine all relevant algorithms. This should be further investigated.

High Gain Fallacy Redirected walking research currently focuses on finding the highest possible thresholds that we can use for manipulation.

However, Section 6.2.3 above shows that high gains fail when they are applied to fine-tuned movements. Future research should investigate how redirection can be employed when precise movements are necessary.

Changes in Virtual Environments Our analysis of the effects that virtual environments have on redirected walking shows that there we can rotate the user more in different environments. We have used linear regression to predict how different environments may compare with each other in this regard. A study that explores immersion in respect to different virtual environments would be welcomed.

Body Tracking Further research should address the user's body pose separately from the orientation of the head. This would improve predictions of motion trajectories, especially in respect to the turn-to-center approach and resetting.

Virtual Reality Sickness Future work should include research concerning how technical factors, such as gains and redirected walking algorithms, affect virtual reality sickness symptoms. This could involve using a modified simulator sickness questionnaire for virtual reality. Such research requires a larger and more varied participant population (see [3] for an overview).

Bibliography

[1] Victoria E. Abraira and David D. Ginty. "The sensory neurons of touch." In: *Neuron* 79 (4 2013). ISSN: 08966273. DOI: 10.1016/j.neuron.2013.07.051.

[2] G. Bruder et al. "Reorientation during body turns." In: *Proceedings of the Joint Virtual Reality Conference of EGVE - The 15th Eurographics Symposium on Virtual Environments, ICAT, EuroVR 2009*. 2009. DOI: 10.2312/EGVE/JVRC09/145-152.

[3] Eunhee Chang, Hyun Taek Kim, and Byounghyun Yoo. "Virtual reality sickness: a review of causes and measurements." In: *International Journal of Human–Computer Interaction* 36.17 (2020), pp. 1658–1682.

[4] Haiwei Chen and Henry Fuchs. "Supporting free walking in a large virtual environment: Imperceptible redirected walking with an immersive distractor." In: *ACM International Conference Proceeding Series*. Vol. Part F1286. Association for Computing Machinery, June 2017. ISBN: 9781450352284. DOI: 10.1145/3095140.3095162.

[5] Timofey Grechkin et al. "Revisiting detection thresholds for redirected walking: Combining translation and curvature gains." In: *Proceedings of the ACM Symposium on Applied Perception, SAP 2016*. Association for Computing Machinery, Inc, July 2016, pp. 113–120. ISBN: 9781450343831. DOI: 10.1145/2931002.2931018.

[6] Pavel Gregoric and Jakob Leth Fink. "Introduction Sense Perception in Aristotle and the Aristotelian Tradition." In: Leiden, The Netherlands: Brill, 2022, pp. 15–39. ISBN: 9789004506077. DOI: https://doi.org/10.1163/9789004506077_003. URL: https://brill.com/view/book/9789004506077/BP000011.xml.

[7] Victoria Interrante, Brian Ries, and Lee Anderson. "Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments." In: *IEEE Symposium on 3D User Interfaces 2007 - Proceedings, 3DUI 2007*. 2007. DOI: 10.1109/3DUI.2007.340791.

[8] H. Iwata et al. "Food simulator: a haptic interface for biting." In: *IEEE Virtual Reality 2004*. 2004, pp. 51–57. DOI: 10.1109/VR.2004.1310055.

[9] Raj Jain. *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling*. eng. New York, 1991.

[10] Luisa Jeffery. *A Filmmaker's Guide to Movie Sound Effects & 6 Iconic Examples*. 2021. URL: https://www.marmosetmusic.com/journal/a-filmmakers-guide-to-movie-sound-effects-6-iconic-examples/ (visited on 12/09/2022).

[11] Jason Jerald et al. "Sensitivity to scene motion for phases of head yaws." In: *APGV 2008 - Proceedings of the Symposium on Applied Perception in Graphics and Visualization*. 2008. DOI: 10.1145/1394281.1394310.

[12] Emil Kjenstad et al. "Experiencing Rotation and Curvature Gain for Redirected Walking in Virtual Reality." In: *International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE '23)*. Vancouver, BC, Canada: ACM, 2023. ISBN: 979-8-4007-0189-4/23/06. DOI: 10.1145/3592834.3592874. URL: https://doi.org/10.1145/3592834.3592874.

[13] Uwe Kloos et al. "Velocity-Dependent Dynamic Curvature Gain for Redirected Walking." In: *IEEE Transactions on Visualization and Computer Graphics* (2012).

[14] Naoya Koizumi et al. "Chewing Jockey: Augmented Food Texture by Using Sound Based on the Cross-Modal Effect." In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. ACE '11. Lisbon, Portugal: Association for Computing Machinery, 2011. ISBN: 9781450308274. DOI: 10.1145/2071423.2071449. URL: https://doi.org/10.1145/2071423.2071449.

[15] Scott A. Kuhl, Sarah H. Creem-Regehr, and William B. Thompson. "Recalibration of rotational locomotion in immersive virtual environments." In: *ACM Transactions on Applied Perception* 5.3 (2008). ISSN: 15443558. DOI: 10.1145/1402236.1402241.

[16] L. N. van der Laan et al. "The first taste is always with the eyes: A meta-analysis on the neural correlates of processing visual food cues." In: *NeuroImage* 55.1 (2011), pp. 296–303. ISSN: 10538119. DOI: 10.1016/j.neuroimage.2010.11.055.

[17] Eike Langbehn et al. "Application of redirected walking in room-scale VR." In: *Proceedings - IEEE Virtual Reality*. 2017. DOI: 10.1109/VR.2017.7892373.

[18] Eike Langbehn et al. "Bending the Curve: Sensitivity to Bending of Curved Paths and Application in Room-Scale VR." In: *IEEE Transactions on Visualization and Computer Graphics* 23 (2017), pp. 1389–1398.

[19] Eike Langbehn et al. "Bending the Curve: Sensitivity to Bending of Curved Paths and Application in Room-Scale VR." In: *IEEE Transactions on Visualization and Computer Graphics* 23.4 (Apr. 2017), pp. 1389–1398. ISSN: 19410506. DOI: 10.1109/TVCG.2017.2657220. URL: https://dl.acm.org/doi/abs/10.1109/TVCG.2017.2657220.

[20] Keigo Matsumoto et al. "Unlimited corridor: Redirected walking techniques using visuo haptic interaction." In: *ACM SIGGRAPH 2016 Emerging Technologies, SIGGRAPH 2016*. 2016. ISBN: 9781450343725. DOI: 10.1145/2929464.2929482. URL: http://dx.doi.org/10.1145/2929464.2929482.

[21] Joseph E. McGrath. "Dilemmatics: The Study of Research Choices and Dilemmas." In: *American Behavioral Scientist* 25.2 (July 1981), pp. 179–210. ISSN: 15523381. DOI: 10.1177/000276428102500205. URL: https://journals.sagepub.com/doi/10.1177/000276428102500205.

[22] Boris Mirkin. "Eleven ways to look at the chi-squared coefficient for contingency tables." In: *American Statistician* 55 (2 2001). ISSN: 00031305. DOI: 10.1198/000313001750358428.

[23] Claire Murphy and William S. Cain. "Taste and olfaction: Independence vs interaction." In: *Physiology and Behavior* 24.3 (1980), pp. 601–605. ISSN: 00319384. DOI: 10.1016/0031-9384(80)90257-7.

[24] Kizashi Nakano et al. "A study on persistence of gan-based vision-induced gustatory manipulation." In: *Electronics (Switzerland)* 10.10 (2021). ISSN: 20799292. DOI: 10.3390/electronics10101157.

[25] Niels Christian Nilsson et al. "15 Years of Research on Redirected Walking in Immersive Virtual Environments." In: *IEEE Computer Graphics and Applications* 38.2 (Mar. 2018), pp. 44–56. ISSN: 02721716. DOI: 10.1109/MCG.2018.111125628.

[26] Niels Christian Nilsson et al. "Estimation of detection thresholds for audiovisual rotation gains." In: *Proceedings - IEEE Virtual Reality*. Vol. 2016-July. 2016, pp. 241–242. ISBN: 9781509008360. DOI: 10.1109/VR.2016.7504743.

[27] Anders Paludan et al. "Disguising rotational gain for redirected walking in virtual reality: Effect of visual density." In: *Proceedings - IEEE Virtual Reality*. Vol. 2016-July. 2016, pp. 259–260. DOI: 10.1109/VR.2016.7504752. URL: https://www.researchgate.net/publication/299469659.

[28] Carl Pfaffmann. *Human sensory reception | Britannica*. 2017. URL: https://www.britannica.com/science/human-sensory-reception (visited on 03/15/2022).

[29] John Prescott. *Multisensory processes in flavour perception and their influence on food choice*. 2015. DOI: 10.1016/j.cofs.2015.02.007.

[30] Nimesha Ranasinghe et al. "Vocktail: A virtual cocktail for pairing digital taste, smell, and color sensations." In: *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*. 2017, pp. 1139–1147. ISBN: 9781450349062. DOI: 10.1145/3123266.3123440.

[31] Sharif Razzaque. "Redirected Walking." AAI3190299. PhD thesis. USA, 2005. ISBN: 0542340534.

[32] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. "Redirected Walking." In: *Proceedings of EUROGRAPHICS* (2001), pp. 289–294. ISSN: 00044172.

[33] Yvonne Rogers. *Interaction design : beyond human-computer interaction*. eng. Place of publication not identified, 2011.

[34] D. Daniel Sheu, Ming Chuan Chiu, and Dimitri Cayard. "The 7 pillars of TRIZ philosophies." In: *Computers and Industrial Engineering* 146 (2020). ISSN: 03608352. DOI: 10.1016/j.cie.2020.106572.

[35] Michael T. Shipley, Matthew Ennis, and Adam C. Puche. "The Olfactory System." In: *Neuroscience in Medicine*. Ed. by P. Michael Conn. Totowa, NJ: Humana Press, 2003, pp. 579–593. ISBN: 978-1-59259-371-2. DOI: 10.1007/978-1-59259-371-2_27. URL: https://doi.org/10.1007/978-1-59259-371-2_27.

[36] Julia Simkus. "ANOVA (Analysis Of Variance): Definition, Types, and Examples." In: *Simply Scholar Ltd* (2022).

[37] B. G. Slocombe, D. A. Carmichael, and J. Simner. "Cross-modal tactile–taste interactions in food evaluations." In: *Neuropsychologia* 88 (2016). ISSN: 18733514. DOI: 10.1016/j.neuropsychologia.2015.07.011.

[38] B.G. Slocombe, D.A. Carmichael, and J. Simner. "Cross-modal tactile–taste interactions in food evaluations." In: *Neuropsychologia* 88 (2016). Special Issue: Synaesthesia and Multisensory Processes, pp. 58–64. ISSN: 0028-3932. DOI: https://doi.org/10.1016/j.neuropsychologia.2015.07.011. URL: https://www.sciencedirect.com/science/article/pii/S0028393215300968.

[39] Vladimír Sojka and Petr Lepšík. "Use of triz, and triz with other tools for process improvement: A literature review." In: *Emerging Science Journal* 4 (5 2020). ISSN: 26109182. DOI: 10.28991/esj-2020-01234.

[40] Jan L. Souman et al. "Walking Straight into Circles." In: *Current Biology* 19.18 (2009), pp. 1538–1542. ISSN: 09609822. DOI: 10.1016/j.cub.2009.07.053.

[41] Charles Spence. *Multisensory Flavor Perception*. 2015. DOI: 10.1016/j.cell.2015.03.007.

[42] Frank Steinicke et al. "Analyses of human sensitivity to redirected walking." In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST*. 2008, pp. 149–156. ISBN: 9781595939517. DOI: 10.1145/1450579.1450611.

[43] Frank Steinicke et al. "Estimation of detection thresholds for redirected walking techniques." In: *IEEE Transactions on Visualization and Computer Graphics* 16.1 (2010). ISSN: 10772626. DOI: 10.1109/TVCG.2009.62.

[44] Richard J. Stevenson, John Prescott, and Robert A. Boakes. "Confusing tastes and smells: How odours can influence the perception of sweet and sour tastes." In: *Chemical Senses* 24.6 (1999), pp. 627–635. ISSN: 0379864X. DOI: 10.1093/chemse/24.6.627.

[45] Qi Sun, Anjul Patney, and Frank Steinicke. "Redirected walking in VR." In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 11900 LNCS. 2020, pp. 285–292. DOI: 10.1007/978-3-030-41816-8_12. URL: http://qisun.me/,https://anjulpatney.com/,https://www.inf.uni-hamburg.de/en/inst/ab/hci/people/steinicke.html.

[46] Qi Sun, Li Yi Wei, and Arie Kaufman. "Mapping virtual and physical reality." In: *ACM Transactions on Graphics*. Vol. 35. 4. 2016. DOI: 10.1145/2897824.2925883.

[47] Alasdair Wilkins. *What is the speed of smell?* Aug. 2011. URL: https://gizmodo.com/what-is-the-speed-of-smell-5831558 (visited on 12/19/2022).

[48] Sarah C. P. Williams. *Human Nose Can Detect a Trillion Smells*. 2014. URL: https://www.science.org/content/article/human-nose-can-detect-trillion-smells (visited on 12/17/2022).

[49] Ruimin Zhang, Bochao Li, and Scott A. Kuhl. "Human sensitivity to dynamic translational gains in head-mounted displays." In: *SUI 2014 - Proceedings of the 2nd ACM Symposium on Spatial User Interaction*. 2014. DOI: 10.1145/2659766.2659783.

Appendix A

Information Relevant to Base Analysis

A.1 Translation Gain

Table A.1 shows the responses of each participant for question 1 in regards to translation gain over the different levels of gain. A value of 0 means that they did not notice the applied redirection, while 1 indicates that they did.

Identification	1	1.1	1.15	1.2	1.25
26317605	0	0	0	0	0
26366743	0	0	0	0	1
26371393	0	0	0	0	0
26371407	0	0	0	0	0
26377648	1	0	0	0	0
26379972	0	0	0	0	0
26380277	0	0	0	0	0
26391338	0	0	0	0	0
26394095	0	0	0	0	0
26395527	0	0	1	1	1
26396376	0	0	0	0	0
26396576	0	0	0	0	0
26396953	0	0	0	0	0
26400567	0	0	0	0	0
26414575	0	0	0	0	0
Mean Noticeability	0.06	0.00	0.07	0.07	0.13

Table A.1: Translation Gain — Raw data (Q1)

Table A.2 shows the responses of each participant for question 2 in regards to translation gain over the different levels of gain. The values indicates a participants quality of experience in virtual reality, where a value of 1 is the lowest rating, while 5 is the highest rating.

Identification	1	1.1	1.15	1.2	1.25
26317605	5	5	5	5	5
26366743	4	5	5	5	3
26371393	5	5	5	5	5
26371407	5	5	5	5	4
26377648	5	5	5	5	5
26379972	5	5	5	5	5
26380277	5	5	5	5	5
26391338	5	5	5	5	5
26394095	5	5	5	5	5
26395527	5	5	4	2	2
26396376	5	5	4	5	5
26396576	5	5	5	5	5
26396953	5	5	5	5	5
26400567	5	5	5	5	5
26414575	5	5	5	4	5
Mean Response	4.94	5.00	4.87	4.73	4.60

Table A.2: Translation Gain — Raw data (Q2)

A.2 Curvature Gain

Table A.3 shows the responses of each participant for question 1 in regards to curvature gain over the different levels of gain. A value of 0 means that they did not notice the applied redirection, while 1 indicates that they did.

Identification	∞	12.5	10	7.5	5
26317605	0	0	0	0	0
26366743	1	1	0	1	0
26371393	0	0	0	1	0
26371407	0	0	1	0	1
26377648	0	0	0	1	0
26379972	0	0	0	0	1
26380277	0	0	0	0	1
26391338	0	0	0	0	1
26394095	0	0	0	1	1
26395527	1	0	0	1	0
26396576	0	0	0	0	0
26396953	0	0	0	0	1
26400567	0	0	0	0	0
26414575	0	0	0	0	1
Mean Noticeability	0.14	0.07	0.07	0.36	0.50

Table A.3: Curvature Gain — Raw data (Q1)

Table A.4 shows the responses of each participant for question 2 in regards to curvature gain over the different levels of gain. The values indicates a participants quality of experience in virtual reality, where a value of 1 is the lowest rating, while 5 is the highest rating.

Identification	∞	12.5	10	7.5	5
26317605	5	5	5	5	5
26366743	4	4	5	3	5
26371393	5	5	5	4	5
26371407	5	5	5	5	4
26377648	5	5	5	5	5
26379972	5	5	5	5	3
26380277	5	5	5	5	3
26391338	5	5	5	5	4
26394095	5	5	5	4	3
26395527	3	5	5	1	5
26396576	5	5	5	5	5
26396953	5	4	4	5	4
26400567	5	5	5	5	4
26414575	5	4	5	4	3
Mean Response	4.79	4.79	4.93	4.36	4.14

Table A.4: Curvature Gain — Raw data (Q2)

As seen in Table A.4, participant #26395527 rated their experience as extremely poor for a curvature level of 7.5, while the remaining participant rated it as fairly high. We thus viewed this participant as an outlier, and checked to see how the quality of experience graph changed without their response. The result of this can be seen in Figure A.1.

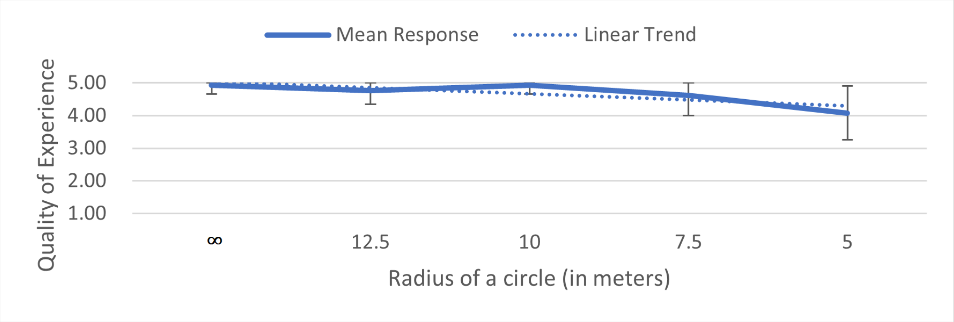


Figure A.1: Curvature Gain — Mean reported quality of experience with outlier removed

A.3 Rotation Gain

Table A.5 shows the responses of each participant for question 1 in regards to rotation gain over the different levels of gain. A value of 0 means that they did not notice the applied redirection, while 1 indicates that they did.

Identification	1	1.1	1.25	1.4	1.55
26317605	0	0	0	0	0
26366743	0	0	0	0	1
26371393	0	0	1	1	1
26371407	0	0	0	0	0
26377648	0	0	0	0	0
26379972	0	0	0	1	0
26380277	0	1	1	0	1
26391338	0	0	0	1	1
26394095	0	1	0	1	1
26395527	0	1	1	1	1
26396376	1	0	0	0.5	0
26396576	0	0	0	1	0
26396953	0	0	1	0	1
26400567	0	0	0	0	0
26414575	0	0	0	0	1
Mean Noticeability	0.07	0.19	0.27	0.44	0.53

Table A.5: Rotation Gain — Raw data (Q1)

Table A.6 shows the responses of each participant for question 2 in regards to rotation gain over the different levels of gain. The values indicates a participants quality of experience in virtual reality, where a value of 1 is the lowest rating, while 5 is the highest rating.

Identification	1	1.1	1.25	1.4	1.55
26317605	5	5	5	5	5
26366743	5	5	5	5	3
26371393	5	5	4	4	3
26371407	5	5	5	5	5
26377648	5	5	5	5	5
26379972	5	5	5	4	5
26380277	4	4	4	5	3
26391338	5	5	5	3	4
26394095	5	3	5	3	1
26395527	5	4	2	1	1
26396376	4	5	5	4.5	5
26396576	5	5	5	4	5
26396953	5	5	3	5	4
26400567	5	5	5	5	5
26414575	5	4	4	5	4
Mean Response	4.87	4.69	4.47	4.25	3.87

Table A.6: Rotation Gain — Raw data (Q2)

As seen in Table A.4, participant #26395527 rated their experience as extremely poor for a rotation level of 1.4, while the remaining participant rated it as fairly high. We thus viewed this participant as an outlier, and checked to see how the quality of experience graph changed without their response. The result of this can be seen in Figure A.2.

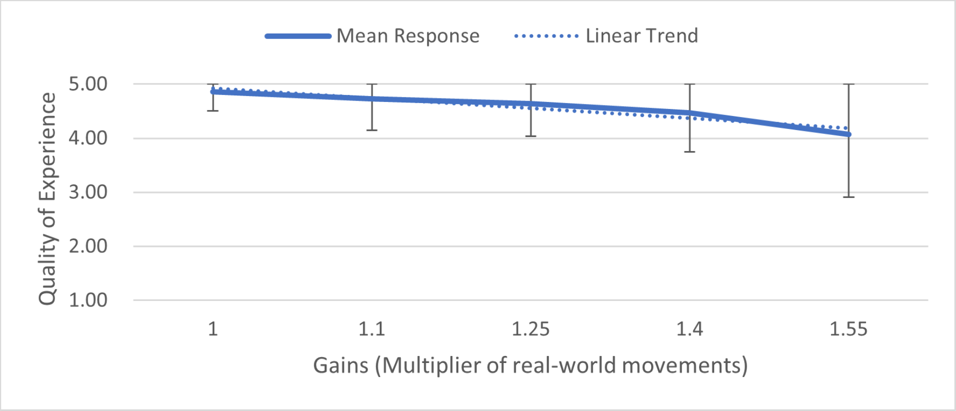


Figure A.2: Rotation Gain — Mean reported quality of experience with outlier removed

Appendix B

Information Relevant to Distractor Analysis

B.1 Rotation Gain with Distractors

Table B.1 shows the responses of each participant for question 1 in regards to rotation gain with distractors over the different levels of gain. A value of 0 means that they did not notice the applied redirection, while 1 indicates that they did.

Identification	1	1.1	1.25	1.5
26317605	0	0	0	0
26371407	0	0	0	0
26377648	0	0	0	0
26379972	0	0	0	0
26380277	0	0	1	1
26391338	0	0	0	0
26394095	1	1	1	1
26396376	0	0	0	1
26396576	0	0	0	0
26512149	1	0	0	1
26512643	0	0	0	0
26530287	1	0	0	1
26532329	0	0	0	0
26532478	0	0	1	1
26552849	1	0	0	0
26552938	0	0	0	0
Mean Noticeability	0.25	0.06	0.19	0.38

Table B.1: Rotation Gain with Distractor — Raw data (Q1)

Table B.2 shows the responses of each participant for question 2 in regards to rotation gain with distractors over the different levels of gain. The values indicates a participants quality of experience in virtual reality, where a value of 1 is the lowest rating, while 5 is the highest rating.

Identification	1	1.1	1.25	1.5
26317605	5	5	5	5
26371407	5	5	5	5
26377648	5	5	5	5
26379972	5	5	5	5
26380277	5	5	3	1
26391338	5	5	5	5
26394095	4	4	4	3
26396376	5	5	5	4
26396576	5	5	5	5
26512149	5	5	5	5
26512643	5	5	5	5
26530287	4	5	5	2
26532329	5	5	5	5
26532478	5	4	5	3
26552849	3	5	4	4
26552938	5	5	5	5
Mean Response	4.75	4.88	4.75	4.19

Table B.2: Rotation Gain with Distractor — Raw data (Q2)

B.2 Rotation Gain without Distractors

Table B.3 shows the responses of each participant for question 1 in regards to rotation gain without distractors over the different levels of gain. A value of 0 means that they did not notice the applied redirection, while 1 indicates that they did.

Identification	1	1.1	1.25	1.5
26317605	0	0	0	0
26371407	0	0	0	0
26377648	0	0	0	1
26379972	0	0	0	0
26380277	0	0	1	1
26391338	0	1	0	1
26394095	0	1	1	1
26396376	0	0	0	0
26396576	0	0	0	0
26512149	0	0	0	1
26512643	0	0	0	0
26530287	0	0	0	1
26532329	0	0	0	0
26532478	0	0	0	1
26552849	0	0	1	0
26552938	0	0	0	0
Mean Noticeability	0.00	0.13	0.19	0.44

Table B.3: Rotation Gain without Distractors — Raw data (Q1)

Table B.4 shows the responses of each participant for question 2 in regards to rotation gain without distractors over the different levels of gain. The values indicates a participants quality of experience in virtual reality, where a value of 1 is the lowest rating, while 5 is the highest rating.

Identification	1	1.1	1.25	1.5
26317605	5	5	5	5
26371407	5	5	5	5
26377648	5	5	5	4
26379972	5	5	5	5
26380277	5	5	3	2
26391338	5	4	5	4
26394095	5	4	3	4
26396376	5	5	5	5
26396576	5	5	5	5
26512149	5	5	5	5
26512643	4	5	5	5
26530287	5	5	5	3
26532329	5	5	5	5
26532478	5	5	4	4
26552849	5	4	3	5
26552938	5	5	5	5
Mean Response	4.94	4.81	4.56	4.44

Table B.4: Rotation Gain without Distractors — Raw data (Q2)

Appendix C

Information Relevant to Virtual Environment Analysis

Figure C.1 shows the raw used to perform a chi-squared test of independence in regards to question 1. The value seen in the table are the mean values after grouping — values close to 0 indicate low noticeability and values close to 1 indicate high noticeability. The sample is comprised of participants that partook in both studies, while the population contains those remaining.

Case	Sample	Population
Fire Drill (1)	0.00	0.00
Fire Drill (1.1)	0.22	0.13
Fire Drill (1.25)	0.22	0.19
Dungeon (1)	0.11	0.07
Dungeon (1.1)	0.20	0.19
Dungeon (1.25)	0.11	0.27

Table C.1: Comparing sample to population — Q1

Figure C.2 shows the raw used to perform a chi-squared test of independence in regards to question 2. The value seen in the table are the mean values after grouping — values close to 1 indicate low quality of experience and values close to 5 indicate high quality of experience. The sample is comprised of participants that partook in both studies, while the population contains those remaining.

Case	Sample	Population
Fire Drill (1)	5.00	4.94
Fire Drill (1.1)	4.78	4.81
Fire Drill (1.25)	4.56	4.56
Dungeon (1)	4.78	4.87
Dungeon (1.1)	4.70	4.69
Dungeon (1.25)	4.89	4.47

Table C.2: Comparing sample to population — Q2