

Andreas Wang

Noticeability and Effectiveness of Distractors in Redirected Walking

Master's thesis in Applied Computer Science

Supervisor: Simon McCallum, Christopher Frantz

June 2019



Norwegian University of
Science and Technology

Noticeability and Effectiveness of Distractors in Redirected Walking

Andreas Wang

01-06-2019

Master's Thesis

Master of Science in Applied Computer Science

30 ECTS

Department of Computer Science

Norwegian University of Science and Technology,

Supervisor: Assoc. Prof. Simon McCallum

Co-Supervisor: Assoc. Prof. Christopher Frantz

Preface

This is a master's thesis in Applied Computer Science which has been written at NTNU Gjøvik during the spring semester of 2019. The project topic came as a result of wanting to apply a prior background of game programming to a field of VR research. Redirected walking was eventually chosen as it consisted of a sub-field of research on distractors which are a perfect fit for VR games that employ redirected walking. This has hopefully provided a more practical and applied perspective to a field of research that otherwise still primarily exists in the theoretical space where smaller technical demos are the largest extent of implementation.

The target audience for this thesis is primarily masters level students in applied computer science and game developers who have an interest in redirected walking. As such, some basic knowledge of how computer games function is expected. Any additional background will be provided for more detailed and specific topics which may not be considered as general knowledge for the target audience.

01-06-2019

Acknowledgement

I want to give special thanks to my supervisors: Simon and Christopher, for their excellent feedback throughout the semester. Their insights and comments have been a great help in reflecting on my work. Furthermore, I would like to thank all of the participants who spent their time to partake in my experiments. Their contributions have provided many interesting results which I could not have achieved otherwise. Finally, I would like to give my thanks to Yijie Zhou for allowing to reuse some of his 3D assets. They have been a great contributor in making Ensemble Retriever look as visually appealing as it does.

A.W.

Abstract

Redirected walking is an area of VR research that aims to optimise the usage of physical tracking space. This optimisation is handled by doing small manipulations to the camera of the user, so they effectively are redirected away from physical walls as they move around in a room scale VR experience. A sub-field within redirected walking is the topic of virtual distractors. These are objects or abstract elements that occupy the attention of the user in a way that facilitates redirection. A problem that current day distractors aim to mitigate is to improve the user's subjective sense of presence and immersion. This problem is handled by fully integrating them into virtual experiences, so they become an essential part of it, rather than some auxiliary tool. Despite this solution, the field is relatively young and current research is currently at the tip of an iceberg in a larger space of inquiry.

In order to increase our understanding of this area of research, this thesis has employed an exploratory approach and developed a VR game titled "Ensemble Retriever", which makes use of state of the art distractors. As part of the work, a new redirection algorithm has been developed: "Align Centre to Future" (AC2F) which uses distractors to align the user's future path towards the centre of their physical space. In conjunction with AC2F, a new reset technique: "Pause - Turn - Centre" has been developed to deal with the various shortcomings of existing reset methods.

By using Ensemble Retriever, two experiments have been conducted to gain deeper insights into various factors on noticeability and effectiveness of distractors in redirected walking. The first of these focused on testing whether there were any significant differences in detection thresholds between two states of Ensemble Retriever: a general walking state and a battle state against an enemy distractor. No significant difference was found for positive rotation gains. Despite these results, it was significantly easier to detect negative rotation gains during distractor battles. Furthermore, an adaptation effect towards positive rotation gains was observed.

The second experiment focused on assessing the effectiveness of the AC2F algorithm. In this case, two conditions were compared: one where the commonly used "Steer to Center" (S2C) algorithm was employed when walking and AC2F when interacting with distractors. The other condition used S2C for everything. The results showed no significant difference between the conditions in terms of the mean number of resets or the mean time needed to align the user's future path towards the centre of the physical space. Despite these results, the first condition resulted in 15.8% fewer failure cases in terms of successful alignments towards physical room centre.

The overall results and discussions around these have yielded a large pool of future work which could be seen as pathways that can be taken to expand our knowledge within the field of distractors and redirected walking as a whole. In addition, Ensemble Retriever and its corresponding source code is openly available for other researchers and developers to see and use.

Contents

Preface	i
Acknowledgement	ii
Abstract	iii
Contents	iv
List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 Topic Covered by The Thesis	1
1.2 Keywords	1
1.3 Problem Description	1
1.4 Justification, Motivation and Benefits	1
1.5 Research Questions	2
1.6 Contributions	2
1.6.1 Research Contributions	2
1.6.2 Secondary Contributions	3
1.7 Thesis Structure	4
2 Related Work and Background	6
2.1 Background: Room Scale Virtual Reality	6
2.2 Redirected Walking, Detection Thresholds, Cybersickness and Presence	6
2.2.1 Background: Redirected Walking	6
2.2.2 Background: Redirection Algorithms	8
2.2.3 Variables That Could Affect Redirected Walking and Detection Thresholds	8
2.2.4 Comfort and Cybersickness	8
2.2.5 Subjective Sense of Presence	10
2.3 Distractors in Redirected Walking	11
2.3.1 Taxonomy of Distractors in Redirected Walking	11
2.4 Usage of Distractors in the Literature	15
2.5 Eye Tracking in Redirected Walking: A New Use Case For Distractors	17
2.6 External Validity of Estimated Detection Thresholds	18
2.6.1 Use of Informed/Uninformed Participants for Redirection	19
2.7 Ethics of Estimating Detection Thresholds	20
2.8 Accessibility of Redirected Walking to Virtual Reality Developers	20
2.9 Choosing a Sufficient Reset Technique	20
2.10 Overall Answers to Research Questions	22

2.11	Usage of Distractors in Other VR Research Fields	23
2.12	Background: Transformation Hierarchies in Game Engines	23
2.13	Background: The Redirected Walking Toolkit	24
2.13.1	How Redirected Walking is Implemented in the Toolkit	24
2.13.2	Toolkit Terminology	24
2.13.3	Toolkit Structure	25
3	General Methods	27
3.1	Literature Acquisition	27
3.2	Personal Data Collection and GDPR Compliance	28
3.2.1	Demographic Questionnaire	28
3.2.2	GDPR Compliance and Data Anonymity	29
3.3	Development Environment	30
3.4	Hardware Environment for Experiments	30
3.5	Software Environment for Experiments	30
4	Implementation	32
4.1	Open Source Repository - GitHub	32
4.1.1	Licensing and Attribution	32
4.2	Redirected Walking Toolkit - Extended Code Architecture	33
4.3	Managing The Extended Architecture	34
4.4	Distractor Enemies	34
4.5	The "Align Centre to Future" Redirector	35
4.5.1	Algorithm Pseudocode	36
4.5.2	Smoothing	37
4.6	The "Pause - Turn - Centre" Resetter	38
4.6.1	Clipping Related Problems	39
4.6.2	Pausing the Game Using "Pause - Turn - Centre"	40
4.7	Experiment Management	40
4.8	Supporting the y-axis in the Redirected Walking Toolkit	40
4.9	Game Design Overview of Ensemble Retriever	41
4.9.1	Virtual Environment	41
4.9.2	Game Flow	43
4.10	Fully Integrating Distractors with Game Mechanics	45
4.10.1	The Contrabass	46
4.10.2	The Oboe	47
4.10.3	The Harpsichord	48
4.10.4	The Violin	48
4.10.5	The Glockenspiel	48
4.11	Employing Context Sensitive Reorientation: Teleporters	50
4.12	Disabling Redirected Walking Towards the End of an Experience	50
4.13	Providing a Distance Magnitude Cooldown on Distractor Triggers	51

5	Experiment 1: Noticeability of Distractors	52
5.1	Method	52
5.1.1	Null Hypothesis	52
5.1.2	Estimating Detection Thresholds	53
5.1.3	Performance Data Collection	55
5.1.4	Participant Sample	55
5.1.5	Experiment Environment/Physical Space	57
5.1.6	Data Post Processing	60
5.2	Results	60
5.2.1	Rotation Detections	60
5.2.2	Curvature Detections	66
5.2.3	Mean Detection Thresholds	66
5.2.4	Test for Normality and Choice of Significance Test	66
5.2.5	Hypothesis Testing	67
5.2.6	Demographic Insights	67
5.2.7	Summary	71
5.3	Discussion	71
5.3.1	Rotation Gain Adaptation Effects	71
5.3.2	Individuality of Detection Thresholds/Events	73
5.3.3	Variability and Asymmetry in Detection Data	73
5.3.4	Curvature Gain Detection Patterns	74
5.3.5	Curvature Gain Adaptation	74
5.3.6	Deciding on Which Estimated Thresholds to Use for Experiment 2	74
5.3.7	Comparing the Results With Fuglestad's Study	75
5.3.8	Effect of AC2F Smoothing on Noticeability	76
5.3.9	Correlation Analysis Discussion	76
6	Experiment 2: Effectiveness of Distractors	77
6.1	Method	77
6.1.1	Hypotheses	77
6.1.2	Data Collection	78
6.1.3	Data Post Processing	78
6.1.4	Changes in Experiment Environment	79
6.1.5	Participant Sample	80
6.1.6	Changes in Ensemble Retriever Between Experiment 1 and 2	82
6.2	Results	84
6.2.1	Relative Effectiveness	84
6.2.2	Alignment Time Effectiveness	87
6.2.3	Supplementary Graphs: Head Rotation for Participant 5	90
6.2.4	Supplementary Graph: Time Taken to Defeat Distractors and Player Baton Level	90
6.2.5	Supplementary Graph: Mean Walking Speed and Prior VR Experience	90

6.2.6	Demographic Insights	90
6.2.7	Summary	94
6.3	Discussion	95
6.3.1	Limitations of Employed Alignment Failure Time Penalty	95
6.3.2	Relationships Between Alignment Fail Rates and Mean Number of Resets	96
6.3.3	Potential Improvements to the Data Processing Approach	98
6.3.4	Minimising the Risk of Cybersickness Buildup With Distractors	99
6.3.5	Success of Employing Mean Detection Threshold Gains From Experiment 1	100
6.3.6	Correlation Matrix Discussion	100
7	General Discussion	102
7.1	Limitations of the Current AC2F Future Path Heuristic	102
7.2	Saliency and Distraction	102
7.3	The Ideal Timing for Switching Redirection Algorithms	103
7.4	Effectiveness of Integration From a Game Design Point of View	104
7.5	Practical Challenges for Distractors in Redirected Walking	105
7.5.1	Movement During Battles	105
7.5.2	Stopping Speeds	105
7.5.3	Concrete Distractors in Confined Spaces	105
7.6	Participant Feedback on Pause - Turn - Centre	106
8	Conclusion and Future Work	107
8.1	Conclusion	107
8.2	Future Work	108
8.2.1	Experiment/Software Specific Future Work	108
8.2.2	Future Work for General Redirected Walking Research	109
8.2.3	The Future of Redirected Walking	111
	Bibliography	112
A	Terminology and Abbreviations	118
B	Reproducing The Experiments	119
B.1	Before Starting	119
B.1.1	Room Setup	119
B.1.2	Hardware Setup	119
B.1.3	Software Setup	119
B.1.4	The Ensemble Retriever Project	120
B.2	Experiment 1	122
B.2.1	Participant Information and Procedure	122
B.2.2	Data Recording/Data Format	123
B.2.3	Data Post Processing	125
B.3	Experiment 2	126
B.3.1	Participant Information and Procedure	127
B.3.2	Data Recording/Data Format	127

B.3.3 Data Post Processing 128

C Demographics Questionnaire 131

D Information Sheets 134

E Approval - NSD 141

List of Figures

1	Illustrated Example of How Redirection Gains Function	7
2	Potential Problem With 2:1 Turn Resetting	21
3	Structure of the Redirected Walking Toolkit	25
4	Extended Structure of the Redirected Walking Toolkit	33
5	Align Centre to Future Algorithm Example	35
6	Pause - Turn - Centre Screenshot	38
7	Pause - Turn - Centre Clipping Bug Screenshot	39
8	Screenshot of the "Hall of The Mountain King"	41
9	Screenshot of the "Hall of The Mountain King" Without the Quiz Wall	42
10	Screenshot of the Environment in Ensemble Retriever	42
11	Top Down Screenshot of Virtual Space That Players Walked Through	43
12	Screenshot of the Tutorial in Ensemble Retriever	44
13	The Distractors of Ensemble Retriever	45
14	The Contrabass Distractor	46
15	The Oboe Distractor	47
16	Sideways 2D Example of Oboe Projectile Path	47
17	The Harpsichord Distractor	48
18	The Violin Distractor	49
19	The Glockenspiel Distractor	49
20	Histogram on Prior VR Experience of Participants in Experiment 1	56
21	Image of Experiment Environment	58
22	Supplementary Image of Experiment Environment	59
23	Top Down Representation of Experiment Environment	59
24	Raw Detection Scatterplot For Rotation Gains	61
25	Finalised Detection Scatterplot For Rotation Gains, Grouped by Algorithm	62
26	Finalised Detection Scatterplot For Rotation Gains, Grouped by Participant ID	62
27	Line Chart of Negative Rotation Gain Detections Between Participants	63
28	Line Chart of Positive Rotation Gain Detections Between Participants	63
29	Scatterplot For Negative Rotation Gain Detections Including Regression Line	64
30	Scatterplot For Positive Rotation Gain Detections Including Regression Line	64
31	Line Chart Showing the Progression of Rotation Gains for Participant 16	65
32	Finalised Detection Scatterplot For Curvature Gains, Grouped by Participant ID	65
33	Boxplot on Positive Rotation Detections in Experiment 1	67
34	Boxplot on Negative Rotation Detections in Experiment 1	68

35	Experiment 1 Demographic Correlation Matrix	68
36	Scatterplot on Rotation Gain Detections in Relation to Prior VR Experience	69
37	Boxplot on Detected Positive Rotation Gains by VR Experience	70
38	Boxplot on Detected Negative Rotation Gains by VR Experience	70
39	Changes in Experiment Environment for Experiment 2	80
40	Histogram on Prior VR Experience of Participants in Experiment 2	81
41	Screenshot of the "Hall of The Mountain King" Post Experiment 1	82
42	Minimum Time Needed To Defeat Distractors Between Participants	84
43	Boxplot of Time Spent Walking Between Conditions in Experiment 2	85
44	Boxplot of Mean Walking Speed Between Conditions in Experiment 2	85
45	Mean Number of Resets Between Conditions for Experiment 2	86
46	Boxplot of Number of Resets Per Participant Between Conditions for Experiment 2	87
47	Boxplot of Time Needed to Defeat Distractor During Failed Alignments	88
48	Histogram on Player Baton Level During Failed Alignments	89
49	Boxplot on Time Needed To Align Participants To Centre (Including Failure Penalties)	89
50	Boxplot on Time Needed To Align Participants To Centre (Successful Cases Only)	90
51	Head Rotation Deltas for Participant 5 in Experiment 2	91
52	Delta Time for Participant 5 in Experiment 2	91
53	Scatterplot Over Time Needed to Defeat Distractors in Experiment 2	92
54	Boxplot on Mean Walking Speed and Prior VR Experience in Experiment 2	92
55	Demographical Correlation Matrix 1 for Experiment 2	93
56	Demographical Correlation Matrix 2 for Experiment 2	94
57	Various S2C Redirection Scenarios During Success/Failure of Centre Alignment	97

List of Tables

1	Variables That Can Affect Detection Thresholds in Redirected Walking	9
2	Distractors in Literature, Framed Within "The Taxonomy of Distractors in Redirected Walking"	16
3	Keywords and Combinations That Were Used for the Literature Search	28
4	Experiment 1: Mean Detection Thresholds	66
5	Experiment 1: Summary Over Contextual Variables in Relation To Detection Thresholds	72

1 Introduction

1.1 Topic Covered by The Thesis

This thesis covers the topic of redirected walking and a particular subarea in this field known as virtual distractors (or simply distractors). A problem with current day room scale solutions in virtual reality is that users often do not have a large amount of space to move around in for these types of experiences. Redirected walking aims to mitigate this problem by unnoticeably redirecting the user while they walk around to create the illusion of a fully explorable virtual world [1]. By doing so, it is possible to make better use of the available physical space while still creating an immersive experience. Despite this optimisation, redirected walking by itself is not sufficient enough to properly redirect the user in smaller physical spaces [2, 3]. By engaging the user with distractors on the other hand, it is possible to increase the degree of unnoticeable redirection while still keeping a high subjective sense of presence [4]. These distractors could be anything from activities in the virtual world to objects that can keep the user's attention.

1.2 Keywords

Virtual Reality, Redirected Walking, Distractor, Distractors, Immersion, Subjective Sense of Presence, Computer Games, Games, Noticeability, Effectiveness, Detection, Detection Threshold, Detection Thresholds

1.3 Problem Description

Redirected walking by itself achieves full effectiveness in very large rooms which are unavailable to most users. As an example: it is necessary to have a room that can fit a circle with a radius of 22 meters to entirely redirect the user in an unnoticeable manner when using one type of redirection technique [2, 3]. It is not only unrealistic to assume that average end-users have access to such large rooms, but also challenging for many modern head-mounted displays to track areas of that size. In smaller physical spaces, the user is expected to be told by the software to reorient themselves a fair amount whenever they are close to the physical walls. These reorientation events can break the user's subjective sense of presence and does not necessarily contribute to an immersive virtual experience.

1.4 Justification, Motivation and Benefits

The limitations of physical space mean that it is all the more important to make sure that reorientation events are as effective and unintrusive to the user as possible [3]. The primary benefits of a good redirected walking solution lie with the end-user as it allows them to experience virtual reality in a more immersive manner as well as providing lower amounts of cybersickness compared

to other forms of locomotion [1, 5]. It further allows the user to walk around in a virtual world that is larger than their available physical space, which could be seen as a crucial part of the experience. It also provides some benefits to the developers of virtual reality (VR) software as they do not have to rely on the limits of physical space to the same degree as they currently do. While there has been some research on the topic of distractors over the years, it is not a large field of research. As such, it would be beneficial to stimulate further research in this field as the problem of limited physical space is unlikely to disappear in the near future. The usage of distractors for the sake of reorientation is particularly useful as they focus on reorienting in a manner that aims to provide lower amounts of intrusion into the experience compared to other approaches.

1.5 Research Questions

While current research on distractors in redirected walking has focused on improving context sensitivity, there are some areas which are left unexplored. One such area is measuring how they affect the noticeability of redirection. Furthermore, if the highest unnoticeable redirection gains with distractors were estimated, it would also be interesting to see how effective the redirection could be in the context of a virtual environment like a game. The reason for using games as a context is that their interactive nature allows to broaden the design space of distractors in a manner that could be more engaging for the user. As such, the following research questions have been established:

RQ₁: How noticeable is redirected walking with distractors in a playful virtual environment?

RQ₂: Given the highest unnoticeable gains, how effective is redirected walking with distractors in a playful virtual environment?

1.6 Contributions

With these research questions in mind, this master thesis has yielded a variety of contributions to the space of redirected walking. The following paragraphs provide a summary of these, categorised by contributions to research and secondary contributions to the redirected walking/VR communities. It should be noted that some background from Chapter 2 may be necessary to understand the finer details of the contributions.

1.6.1 Research Contributions

For redirected walking research, this thesis has provided the following contributions:

The Taxonomy of Distractors in Redirected Walking

As part of the literature review in Chapter 2, a taxonomy detailing components and elements of distractors seen in the literature has been generated. The taxonomy provides an empirically-supported classification, discussion points and analysis for the research and development around distractors while allowing for extensions by future work.

Providing New Insights With Exploratory Methods

Given that the research field on distractors is still relatively small, this study has made use of rather exploratory approaches and methods. These methods have resulted in a large amount of potential future work for researchers as well as insights which might not have been possible to find with pre-established methods. Specifically for RQ_1 , an incremental detection threshold estimation method has been developed with inspiration from Fuglestad's research [6]. For RQ_2 , a new effectiveness metric has been introduced, which consists of comparing the time taken for a distractor to align the user towards its goal relative to its active time.

In general, the developed experiment environment can be considered as having a larger scope than existing work. This scope means that there are far more potential variables in play as the scenario is closer to the real world and as such, less controlled. As a result, it does create some challenges in terms of mitigating the influence of potentially unaccounted variables. By taking this approach though, it is possible to gain insights and find questions which might not have been possible otherwise. There has also been a significant focus on reproducibility to ensure that the exploratory methods that have been applied can be reproduced and reused by future research.

Experiment Results

As part of providing answers to RQ_1 and RQ_2 , two experiments have been conducted. The results from these experiments are considered as another contribution. The first of the experiments compared the differences in noticeability between two states: a walking state and a distractor battle state. No significant difference was found between the two for positive rotation gains. Despite this, it was significantly easier to notice negative rotation gains in the battle state. Furthermore, an adaptation effect towards positive rotation gains was observed.

The second experiment compared two effectiveness metrics between two separate conditions. The first of these consisted of using the "Steer to Center" (S2C) algorithm while walking and the developed "Align Centre to Future" (AC2F) algorithm during distractor battles. The second condition consisted of using S2C for both walking and battle states. No significant difference was found in terms of the mean number of resets or time taken to reach alignment towards physical room centre. Despite these results, it should be noted that the first condition experienced 15.8% fewer failure cases in terms of completing alignment before the distractor was defeated.

1.6.2 Secondary Contributions

Outside of the research contributions, there are a variety of secondary contributions which could be considered as beneficial to developers and researchers alike. These include:

Providing an Example of State of the Art Distractor Usage

The thesis has provided an example of how noticeable redirected walking is when using state of the art distractors. The reason for mentioning it has provided an example is that the design space for distractors is vast and as such, some implementations might be more effective than others. As more and more research into the field is generated, it is possible to improve the understanding of the effect that various variables might have on the results.

Contributing to a Young Field of Research

Furthermore, the thesis has contributed to a field of research that currently is reasonably small and young. Due to the size and age of this field, it is beneficial to provide additional results that can point towards the effectiveness and noticeability of redirected walking with distractors. By doing so, it is possible to validate existing research as well as providing data that can be used for consideration by other researchers with interest in the topic.

An Openly Available VR Game: "Ensemble Retriever"

This thesis has also contributed an openly available VR game, titled "Ensemble Retriever" which makes use of redirected walking with distractors. The developed game has been used for two experiments to provide answers to the previously established research questions. The scope of the game can be considered as larger and more complex than existing work, making it a valuable example of redirected walking integration in larger projects. In addition, a new redirection algorithm, as well as a new resetting technique has been developed for this thesis. Further information on these can be found in [Chapter 4](#).

1.7 Thesis Structure

The thesis itself has a nested structure which corresponds to each research question. Each research question has an experiment dedicated to it and includes its own method, result and discussion components. The reasoning for this structure is that parts of the results for the experiment on noticeability are prerequisites for the second experiment, which focuses on effectiveness. The overall structure of this thesis is as follows:

1. Introduction
 - This chapter is an introduction to the topic, problem space, associated research questions and contributions.
2. Related Work and Background
 - This chapter consists of a literature review which has been written with the established research questions in mind. It also provides the necessary background for understanding redirected walking.
3. General Methods
 - This chapter details general methods which apply to the thesis as a whole.
4. Implementation of Ensemble Retriever
 - This chapter provides abstracted implementation details for the developed Ensemble Retriever game, and its corresponding redirected walking functionality. The overarching game design of the game and its usage of distractors is also detailed.
5. Experiment 1: Measuring the Noticeability of Distractors

1. This chapter is focused on an experiment which tests for differences in noticeability between a walking state and a distractor battle state when playing Ensemble Retriever.
6. Experiment 2: Measuring the Effectiveness of Distractors
 1. This chapter is focused on an experiment that compares effectiveness metrics in terms of the mean number of resets and time taken for alignment towards physical room centre. This comparison is made between a control condition and an experimental one that employs a newly developed redirection algorithm.
7. Overarching Discussion
 - This chapter takes a general look at the thesis as a whole and discusses various limitations, future improvements and challenges which have been observed throughout this research.
8. Conclusion and Future Work
 - The thesis then concludes on the research that has been conducted and ends with a large amount of future work which would benefit from further exploration.

2 Related Work and Background

In order to see which extent the research questions have been answered by existing work, a literature review has been conducted in this chapter. The sampling procedure of the literature for this review can be found in Chapter 3. This review is an extension from a previously written literature review in the IMT4205 Research Project Planning course.

2.1 Background: Room Scale Virtual Reality

Before discussion and information around redirected walking can take place, it is first necessary to understand what room scale VR is. When it comes to virtual reality, there are two primary modes which are used: a seated mode and a room scale mode. A seated mode mainly makes use of full rotation tracking on all three axes, while the physical position of the user is not tracked. In room scale VR, both rotation and physical position are tracked. This tracking mode allows the user to move around in their physical space while maintaining a 1:1 ratio of movement and interaction with the virtual world using fully tracked controllers. Redirected walking functions within the room scale mode of VR as this is the only mode where redirection of physical movement is possible.

2.2 Redirected Walking, Detection Thresholds, Cybersickness and Presence

Before reviewing the literature on distractors, it would be beneficial to first review the general topic of redirected walking. This section provides some background as well as relevant existing work within this area.

2.2.1 Background: Redirected Walking

The concept of redirected walking was originally presented by Razzaque et al. [1] as an alternative to real walking in virtual environments. The primary motivator behind its introduction was to optimise the usage of physical tracking space. This optimisation, in turn, allows for the development of virtual environments that are larger than the physical tracking space at what was considered a minimal increase in simulator sickness.

Since its introduction, redirected walking has seen a fair amount of development as an area of research. One particularly important study was Steinicke et al.'s research, which formalised the concept of detection thresholds and introduced a taxonomy of redirected walking techniques [2]. As part of the taxonomy, they introduced the concept of three types of redirection gains:

- Translation Gain
- Rotation Gain
- Curvature Gain

Translation gain is defined as a gain of translational movement in the virtual world compared

to the real world. Rotation gain is defined as a gain of head rotation in the virtual world compared to the real world. These rotational gains are usually applied on the vertical axis. Finally, curvature gains are defined as a camera manipulation that constantly injects small changes in vertical angles as the user walks around. This injection allows the user to be redirected so they physically walk on a curve when it appears that they walk in a straight line virtually. Curvature gains are defined as the ratio: $\frac{1}{r}$ where r is the radius of the circular arc that the user is walking on. Some direct examples of how the three gains work could be as follows:

Translation Gain of 2: The user walks 5 meters in the real world but travels 10 meters virtually.

Rotation Gain of 2: The user rotates 180 degrees in the real world but rotates 360 degrees virtually.

Curvature gain of 1 (radius = 1m): The user has travelled on a quarter circle after $\frac{\pi}{2}$ meters in the real world while walking in a straight line virtually.

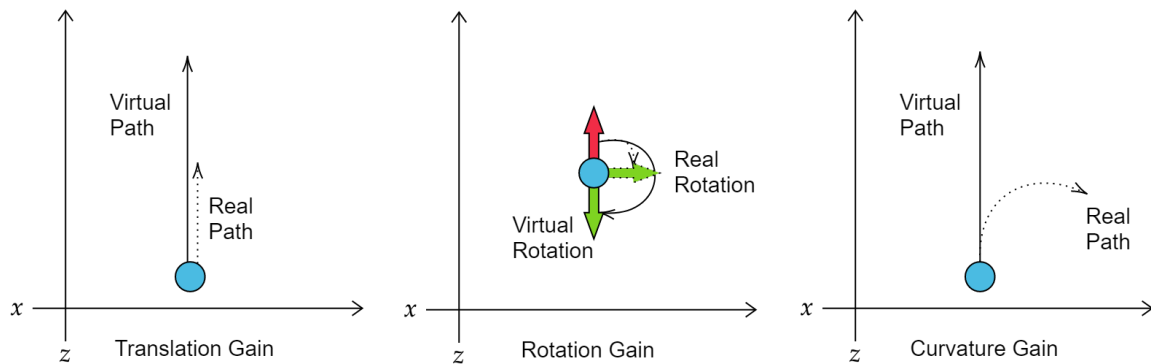


Figure 1: This illustrated example shows how the three primary gains in redirected walking function.

Further illustration on how the different gains work can be seen in Figure 1.

Detection thresholds allow for the estimation of gains that can be applied without the user noticing them. As such, it has become a core of many other studies in the field. The taxonomy itself has since been extended by Suma et al. [7] to provide a more comprehensive look into additional redirection techniques. Outside of the three established redirection gains, an additional fourth type has recently been proposed by Langbehn et al. [8]. Their study presents the concept of bending gains which are similar to curvature gains but only applied whenever the user walks on a curve in the virtual world.

In terms of relevance to research questions, the estimation of detection thresholds is directly relevant to RQ_1 as it provides a means to find undetectable gains. Despite this, there are some problems with this method of estimation that make it rather unsuitable to use. These are further mentioned in Section 2.4.

2.2.2 Background: Redirection Algorithms

In order to redirect a user, it is necessary to have a redirection algorithm. There are a variety of different redirection algorithms, but the two most common and generalised ones are as follows:

"**Steer-to-Center**"(S2C): A redirection algorithm that aims to steer the user towards the centre of the physical tracking space.

"**Steer-to-Orbit**"(S2O): A redirection algorithm that aims to steer the user so they walk on the edge of a circle that encompasses the physical tracking space.

Both of these algorithms were originally presented by Razzaque [9] and have since been improved by Hodgson et al. [10]. The basis for these algorithms is to make use of rotation, curvature and translation gains in a manner that steers the user towards a desired point or direction. Among the two algorithms, S2C is generally believed to have the best performance, but S2O can perform better when long straight paths are travelled [10]. S2C has also been demonstrated to work particularly well for smaller physical spaces which is common for most virtual reality consumers today [3].

Among the various redirection algorithms that exist, the most relevant for this thesis is Peck et al.'s "Improved Redirection with Distractors (IRD)" [11]. This algorithm is an extension of the S2C algorithm with one primary addition: it makes use of virtual distractors to reorient the user towards the centre of the physical space when approaching physical walls. The usage of virtual distractors in redirected walking is further detailed in Section 2.3.

2.2.3 Variables That Could Affect Redirected Walking and Detection Thresholds

One thing to note about detection thresholds and the efficiency of redirected walking is that there are a variety of variables that can impact these. All the potential variables that were found throughout the sample of literature can be found in Table 1. Since each variable is only briefly presented, there is a fair amount of abbreviated and potentially new terminology in use. The usage of abbreviated terminology will also increase from this point onwards in the thesis. As such, a short description of these can be found in Appendix A for quick referencing.

The variables that are found in Table 1 are relevant to this thesis as they can be used to inform the design of the virtual test environment, the use of redirection algorithms depending on available physical space and how to potentially maximise undetectable gains. Furthermore, the study by Azmandian et al. [3] provides a means to measure the quality of redirection which is relevant for RQ_2 .

2.2.4 Comfort and Cybersickness

While redirected walking aims at optimising the use of available tracking space and still leverage the benefits of real walking, it can result in some problems. High redirection gains tend to result in cybersickness, which can be seen as a form of motion sickness [23]. In doing so, one of the primary benefits of real walking is lost. Ideally, the increase in cybersickness from redirected walking should be minimised to provide the best user experience while still efficiently using the tracked space. In relation to detection thresholds, Fuglestad's research has shown that there might be an additional

Variable	Research Discussing Variable	Research Results Comments in Parentheses
Size + Shape of Physical Tracking Space	[3]	There is no single "optimal" size. Square shaped tracking spaces are the best choice for S2C and S2O algorithms. S2C performs best in tracking spaces under 15m x 15m.
Optical Flow / Visual Density in VE	[12, 13, 14, 15]	Virtual environment size has no significant effect on detection thresholds. It appears that low visual density/optical flow could make it harder to notice redirection. Textures and global illumination have no significant effect on translation gain noticeability.
Hardware: HMD Field of View	[6] Potentially relevant: [16]	Detection thresholds for rotation and translation gain application are significantly lower with modern day hardware. This could be caused by an increase in HMD field of view. (This might correlate with increased optical flow).
Speed of Walking	[17]	Likelihood of detecting curvature gains is significantly lower when walking slower. Dynamic curvature gains allow for larger travel distances between resets compared to static gains.
Engagement / Distraction	[2, 18, 19] Potentially relevant: [16, 17]	Whenever a user is engaged with a primary task or distracted by something, they appear to be less likely to notice that redirection is applied.
Awareness of Redirection	[2]	If the user is not aware of the use of redirection, higher gains can be applied without being noticed.
Gender	[20]	Men seem to be more sensitive to curvature gains than women.
Adaptation: Curvature Gains	[2, 21, 22]	Exposing users to curvature gains for 20 minutes makes it harder to detect the gain. Similar effects might be possible for rotation, translation and bending gains. Gradually increasing the strength of curvature gains appears to cause some adaptation.

Table 1: Variables that can affect detection thresholds in redirected walking.

threshold between noticeable redirection and increases in cybersickness [6]. These insights mean that it could be feasible to use estimated detection threshold values for redirection with limited risk of cybersickness increases, even if the user notices it at times. Similar results have been found by Rietzler et al. [24], suggesting that the use of curvature gains can be noticeable, but still acceptable.

Outside of high redirection gains, there are also other factors that could increase cybersickness or limit user comfort. Dynamic field of view has for example been shown to potentially increase cybersickness [16]. Women might also be slightly more susceptible to cybersickness than men due

to a slightly larger field of view [25]. Newer results from Nguyen et al. suggest otherwise though, as no significant increase in simulator sickness was found between genders in one of their experiments [20].

Due to the potential negative implications of standard redirected walking techniques, some researchers have developed new means of redirection. Suma et al. have made use of change blindness as a way to redirect without any type of gains [26]. Despite this, their method only works for indoor environments and requires additional engineering for each individual room amongst these. Sra et al. have made use of scene rotation whenever the user is engaged with a task to leverage inattentive blindness [19], but their approach only works with predefined paths.

From the point of view of a virtual environment designer, it would not be ideal to impose too many restrictions on how the environment is designed. At the same time, it is crucial to consider that cybersickness should be minimised as much as possible. Not doing so, can result in some ethical ramifications which are discussed in Section 2.7. For this thesis in particular, it is helpful to know that cybersickness can exist at a higher threshold than detection as this means that estimated gains could be possible to use safely. This knowledge has also been used to inform the method for estimating detection thresholds.

2.2.5 Subjective Sense of Presence

Subjective sense of presence is a central part of virtual reality experiences. By properly immersing the user into a virtual world, it is possible to provide an engaging user experience. Subjective sense of presence can be negatively affected by a variety of techniques in redirected walking, which can compromise the overall experience of the user.

The first of these are high redirection gains. Similar to Fuglestad's research that shows a difference between noticeability and cybersickness thresholds [6], Schmitz et al. presented that presence/immersion breaks at a different threshold from detection [18]. This insight means that certain gains of redirection can be noticed without resulting in cybersickness increases or breaks in presence. Outside of redirection gains, another technique that can affect the subjective sense of presence is forced reorientation/resetting. Resetting techniques are fail-safes that are used whenever the user starts to exit the physical tracked space due to insufficient redirection [27]. An example of this would be to instruct the user to stop and rotate 360 degrees in the virtual world while they only rotate 180 degrees in reality due to applied rotational gains [7]. By doing so, the user is reoriented so that they no longer are in danger of leaving the tracked space. The problem with these types of techniques is that they are very intrusive and easily break any subjective sense of presence as users have to temporarily stop their in-game activity prior to resetting.

Azmandian et al. mention that the average user should expect to have a physical tracking space that is 10m x 10m or lower, which results in many resets as unnoticeable redirection is not sufficient [3]. Therefore, they suggest that the focus should be on improving existing reset/reorientation mechanisms and improving integration into the experience to limit breaks in presence. Improved subjective sense of presence and improved redirection are among the main areas that distractors aim to improve [4, 5] which are the focus of this thesis.

2.3 Distractors in Redirected Walking

Distractors in redirected walking were originally presented by Peck et al. in 2009 [4]. In their study, the participants were instructed to watch a moving sphere and use this "distraction" as a means to increase redirection. Since then, various researchers have further improved distractors, although the term itself has acquired a few different semantic meanings. At the simplest level, a distractor when employed with redirected walking, aims to occupy the user's attention so that it is harder to notice redirection. As such, it is also possible to increase redirection during distraction [2]. This effect could potentially be a result of inattention blindness which Sra et al. mention in their study on distractors [19]. Distractors are triggered in a similar way to previous reorientation techniques (often shortened to "ROT", "ROTs" or simply resets), meaning that they activate whenever the user moves towards the edge of the physical tracking space. Compared to previous ROTs, distractors activate at a lower distance from the centre which means that existing ROTs can still be used as a fail-safe if the distractor itself fails to redirect the user [7]. Results from previous research also suggest that distractors result in higher levels of subjective presence compared to previous ROTs [5].

From the acquired sample of literature, it does not seem that there is any formal taxonomy that defines the elements of distractors or their types. The closest to this would be Suma et al.'s taxonomy on redirection techniques [7], but this taxonomy is too general to specify the details for distractors. As such, a grounded theory approach [28] has been used to generate a taxonomy that classifies the most apparent elements that distractors consist of. This taxonomy will also be used as a framework to review the acquired sample of literature on distractors.

2.3.1 Taxonomy of Distractors in Redirected Walking

In cinematic VR, a similar topic to distractors is guided attention. While the goals behind a distractor and guided attention are somewhat different, there is some overlap that could be useful to consider. Taking inspiration from the taxonomy on redirected walking by Suma et al. [7], Nielsen et al. have created a taxonomy of cues for guiding user attention in VR [29]. Their taxonomy consists of three dimensions: explicit/implicit cues, diegetic/non-diegetic cues and whether they limit the ability to interact with the virtual environment (shortened to VE). Limiting interaction in the VE is not quite as relevant for redirected walking with distractors as physical walking is required, but the other two dimensions have been adapted into this taxonomy. The following sections describe general elements that can apply to any distractors, a distinction between two types of distractors and some elements that one of these can consist of.

Explicitness

One of the relevant elements from Nielsen et al.'s taxonomy is explicitness. This element is split into explicit and implicit cues. An explicit cue consists of communicating that an event or object is deserving of attention while an implicit cue is meant to guide attention by simply being salient or interesting. In terms of distractors, these can also be explicit or implicit. An example of an explicit distractor would be a moving enemy in a VR game or something that the user has been told to pay attention to whenever it appears. As long as the user is told that they should pay attention to

something, regardless of its purpose in the VE we can consider it as an explicit distractor.

Implicit distractors on the other hand, would be distractors that can catch the user's attention in an almost instinctive manner. These could be salient elements that pop up in the peripheral vision of the user, potentially making them turn their head to see what it is. Another example would be a firefly that flies around and could catch the user's attention simply due to its salience in a darker environment. Compared to an explicit distractor like an enemy that the user knows they have to defeat, implicit distractors do not explicitly communicate how the user should react or interact with them.

Currently, the usage of distractors is mostly of the explicit variety. A reason for this could be that the risk of ignoring an implicit distractor might be high. Unless the implicit distractor intrudes in a way that requires action from the user, it might not be seen as anything other than a detail in the scenery. The risk of ignoring a distractor can be problematic as there might not be enough redirection to avoid moving to the physical tracking boundaries. An effective implicit distractor might also be more challenging to design. It could be beneficial to have some background in psychology or neuroscience to understand what a natural user response would be to various implicit distractor scenarios.

Context Sensitivity

An important aspect of distractors in the state of the art literature is context sensitivity [30, 31, 19]. Initial distractor implementations were fairly generic and worked similarly to standard ROTs, albeit with some changes. An example of this would be the hummingbird distractor from a study by Peck et al. [5]. This distractor appears whenever the user approaches the physical boundaries and flies back and forth in front of the user. The user has in this case been instructed to keep their attention on the hummingbird, which then is exploited for redirection and reorientation. The downside of these generic types of distractors is that they serve no other purpose in the virtual experience other than to exploit the user for redirection and reorientation. This context insensitivity might not be ideal in terms of subjective sense of presence and can be seen as too repetitive if used too frequently.

Instead, the focus with state of the art distractors has been to integrate them into the virtual experience so they serve additional purposes. VR games in particular are a good fit for integrating distractors as they can be included as game mechanics that are a central part of the experience. By doing so, there is less of a need for specialised instruction outside of understanding the premise of the game. There is also a belief by researchers that doing so will increase the subjective sense of presence [31, 19]. Improving the subjective sense of presence can be an important consideration in the design of distractors as they could improve the user experience.

Based on what has been seen in the literature, distractors could be categorised within four types of context sensitivity: Insensitive, Visually Integrated, Mechanically Integrated and Fully Integrated (Both visually and mechanically).

Distractor Types

The term "distractor" has primarily been used to describe objects in the virtual environment (VE) that behave in a way that allows for redirection to be applied during distraction. A study by Sra

et al. [19] made use of distractor activities or simply "attractors" as they called it, which carries a slightly different semantic meaning. Due to this, it could be useful to differentiate between these two by splitting them into concrete and abstract categories.

Concrete Distractors

If a distractor is an object or virtual existence in the VE, we can consider this as a concrete distractor. Concrete distractors consist of a variety of design elements which are further specified in the section following the description of abstract distractors.

Abstract Distractors

An activity on the other hand, can be considered as an abstract distractor. An activity can keep the user's attention by itself, but it can also consist of concrete distractors which require further attention of the user. If an abstract distractor is defined as an activity, then it is arguably explicit by nature as the user generally is aware of how they have to engage with it. An example of an abstract distractor would be to play a game, as various game elements and mechanics can keep the attention of the user. It does not matter what type of game the user plays as simply being engaged with it can be considered as a distraction, although its strength can of course vary.

Abstract distractors can also be context sensitive or insensitive. An example of a context insensitive, abstract distractor would be to ask the user to perform a task that is entirely unrelated to what they are doing in the VE. An example of a context-sensitive, abstract distractor could be the activity of stargazing in an exploration-focused night-time scene. The activity of looking up into the sky could in this case allow for scene rotation while the user is looking upwards. The key to context sensitivity for these types of distractors is that they are believable and natural to the environment they are used in.

Elements of Concrete Distractors

Concrete distractors can consist of a variety of elements that might result in specific user behaviour or to improve the user experience. The following paragraphs discuss the most apparent elements that were found by reading through the acquired sample of literature on distractors. This part of the taxonomy should be very easily extendable so that additional elements can be added in future work.

Diegetic/Non-diegetic Existence

The second element from Nielsen et al.'s framework regarding diegetic/non-diegetic cues is mostly relevant to concrete distractors. They define a diegetic cue as a cue that is a part of the world in the VE. By being part of the world, these cues are not only visible to one user, but also to any other users or computer controlled characters. An example of this would be to have a nearby volcano erupt. This event is then noticeable by both the players and any potential virtual characters in the narrative. A non-diegetic cue on the other hand, is only visible to one user. An example of this would be a HUD or GUI that only one user sees. To provide some examples within the scope of concrete distractors: a concrete, diegetic distractor could be a virtual character that the player interacts with while a concrete, non-diegetic distractor could be useful information in a HUD that the player wants

to frequently keep attention on.

Diegetic cues could potentially improve the subjective sense of presence as Nielsen et al. achieved borderline significant results in their study [29]. This insight could in turn also be relevant for concrete distractors. At the same time, in the context of games, it is not uncommon to have certain UI or elements that always are visible on the screen. If the game does not aim for realism, suspension of disbelief from the player could still result in a high subjective sense of presence. Due to this, one should not outright discredit the usage of distractors that are non-diegetic.

Movement

Another element that concrete distractors consist of is movement or the lack of it. In terms of redirection, a concrete distractor can move around in a way makes the user turn their head. This movement can be used to apply rotational gains. An example of this would be a concrete distractor that orbits around the user and is important enough that they want to keep it in their vision at all times. If a concrete distractor moves towards the user, it might result in the user attempting to avoid it which can be used to move them away from physical walls or to apply redirection. A concrete distractor can also be useful for head-turning when static. An example would be to place a treasure chest at an angle from the user. In order to get the treasure, the user has to turn their head and move towards it which can be combined with rotation, curvature and translation gains.

First Appearance in Vision

How the concrete distractor first appears in the user's vision is also something to consider. It could appear in plain sight, in the peripheral vision or outside of the user's vision. If a concrete distractor appears in the user's peripheral vision, it might catch their attention and make them turn their head to see what it is. If it appears outside of the user's vision, an audio cue or visual effects could be used to direct their attention towards where it is. If it appears in plain sight, the distractor might need to move around in order to apply rotational gains or to keep the user occupied so they do not notice a scene rotation. An example of this would be a merchant that allows the user to view their wares in a book or pamphlet. While the user is reading, the scene can rotate for the sake of redirection.

Deterrents and Attractors

Peck et al. introduced the concept of deterrents as a supplement to distractors [5]. In their study, a deterrent was defined as something that deters the user from moving towards it. An example of a deterrent would be walls of fire that a fire-breathing dragon creates in Chen and Fuch's study [31]. In their study, these deterrents were used as a means to deter the user away from physical walls through strategic placement. Given that deterrents are capable of catching the user's attention, they could be seen as a type of concrete distractors.

The opposite of a deterrent would be an attractor which the user wants to engage with. This definition is similar to Sra et al.'s definition of attractors [19], but instead of an activity, this definition is focused on concrete distractors. One of the previous examples of a treasure chest appearing at an angle that forces redirection could also be considered as an attractor.

Saliency

The final element of concrete distractors is saliency. Saliency is defined as how easily a visual object stands out from its surroundings. In terms of distractors, saliency can be important so that the user can quickly identify and place their attention on any concrete distractors that are used. An example of salient things that are effective at drawing attention could be moving objects or moving lights [32]. Visual saliency is comprehensive enough to be its own topic, although Nielsen et al. have mentioned some central factors that are believed to influence saliency [29]:

- Luminance Contrast
- Edge or Line Orientation
- Colour
- Motion
- Stereo Disparity

One thing to note about these factors is that it is also important to consider how they contrast with the surrounding environment. Furthermore, the distribution of salient regions in a scene has a significant impact on how the user explores it as seen in a study by Sitzmann et al. [33]. Their findings show that having few salient regions in a scene results in attention being shifted faster towards anything salient. As an additional effect, this attention is also more concentrated. These insights have been considered in the design of the virtual experience that is used for Experiment 1 and 2 in this thesis.

2.4 Usage of Distractors in the Literature

The previously generated taxonomy has been used as a framework to map out the usage of distractors in the acquired sample of literature. This mapping can be found in Table 2.

When looking at the current usage of distractors from the sample of literature, a few insights can be gained. Most distractors are explicit and the only instances that could be regarded as implicit were in cases of deterrents. Due to this, there are potentially unexplored areas when it comes to designing implicit distractors. As already mentioned in the description of the taxonomy, these can be rather challenging to design which is why we might not see them being used very often. Despite this, recent developments into redirected walking with eye tracking hardware makes use of techniques that closely resemble implicit distractors and could arguably be categorised among these. Some additional information on this can be found in the following section.

All of the concrete distractors in this review were also diegetic. This means that there is room to explore non-diegetic, concrete distractors that are integrated into the experience. Non-diegetic distractors could for example be used as aids for the user in games or scenarios where extra contextual information is helpful.

Outside of the previously discussed elements, it should be noted that initial forms of distractors were mostly insensitive to context and could be seen as relatively generic. Over time, distractors have started to become more visually integrated with their respective virtual environments. The current state of the art has focused on fully integrating distractors by combining them with game

Distractor	Used in Study	Explicitness	Context Sensitivity	Distractor Type	Additional Relevant Elements
Floating Sphere	[4]	Explicit	Insensitive	Concrete	Non-diegetic, Has movement, Appears in front of user.
Butterfly	[4, 16]	Explicit	Insensitive: [4], Visually Integrated: [16]	Concrete	Diegetic, Has Movement.
Ghost	[11]	Explicit	Visually Integrated	Concrete	Diegetic, Has Movement, Appears in front of user.
Hummingbird	[5, 7]	Explicit	Insensitive	Concrete	Diegetic, Has Movement, Appears in front of user.
Horizontal Bars	[5]	Implicit	Fully Integrated	Concrete	Diegetic, Static, Fades into view as user nears physical walls, Used as a deterrent.
Temporary Objectives	[34]	Explicit	Undefined	Abstract	None.
Moving NPCs	[17]	Implicit	Fully Integrated	Concrete	Diegetic, Has movement, Moves into view from the side or walks in front of the user, Used as a deterrent.
Flying Dragon	[30, 31]	Explicit	Fully Integrated	Concrete	Diegetic, Has movement, Moves towards user from current position when active, Can be considered as a attractor.
Flame Walls	[30, 31]	Implicit	Fully Integrated	Concrete	Diegetic, Static, Spawned by dragon when near physical walls, Used as a deterrent.
Watching Exotic Birds	[19]	Explicit	Fully Integrated	Abstract	The activity itself consists of concrete distractors (birds). These birds move across the sky and have to be viewed through a pair of binoculars.
Observing Insects	[19]	Explicit	Fully Integrated	Abstract	The activity itself consists of concrete distractors (insects). Among the many insects, the user is asked to keep one of them in focus.
Observing Insect in a Piece of Amber	[19]	Explicit	Fully Integrated	Abstract	The user has to hold a piece of amber towards the sky and rotate so they can find a viewing angle that allows them to see the insect inside.
Interacting with NPCs	[19]	Explicit	Fully Integrated	Abstract	The user has to hold a conversation with a NPC which acts as a concrete distractor.
Playing a target shooting minigame	[6]	Explicit	Fully Integrated	Abstract	The targets that move around could be considered as concrete distractors.

Table 2: List of all distractors that were used in the sample of literature, framed within the "The Taxonomy of Distractors in Redirected Walking".

mechanics and activities. Even so, the integration could go further as the currently developed prototypes are very simple. This thesis has aimed to develop a game prototype with a slightly broader scope than existing work and fully integrate it with distractors. The overall design space of fully integrated distractors is relatively large, which allows for additional exploration by researchers. There is not much documentation on the effectiveness of these types of distractors either, creating an opportunity for new research in the area [35]. This thesis aims to provide some documentation on this through an experiment related to RQ_2 .

The significant focus on context sensitivity in the field of distractors does create somewhat of a mismatch with the rigid structure of Steinicke et al.'s detection threshold estimation procedure [2] though. In particular, it becomes hard to create a virtual experience where distractors are fully integrated as the method for estimating detection thresholds would interfere too much with the possible design space of the experience. This issue is further coupled with some ethical problems which are discussed in Section 2.7. In the end, these issues makes it hard to justify using the standard procedure for estimating detection thresholds. As such, an alternative procedure is employed which is detailed in Section 5.1.2.

2.5 Eye Tracking in Redirected Walking: A New Use Case For Distractors

A sub-field within redirected walking that recently has seen some major development is the employment of eye tracking hardware to further redirect users. Sun et al. have created a redirection system that performs small redirections during eye saccades, which consist of a small moment of temporary blindness [36]. Furthermore, they make use of a subtle gaze direction method which adds additional salience to nearby objects or elements in the environment that are in the peripheral vision. As a result, this implicitly encourages the eye to move towards these regions. This technique could very well be framed as using implicit distractors from the previously presented taxonomy and provides an interesting use case for distractors. By combining distractors with state of the art redirection methods that use eye tracking, it could be possible to further improve the potential strength of redirection. As distractors are aimed at keeping user attention, they would be very well fit to help with triggering eye saccades, particularly so if they move around.

Outside of Sun et al.'s study, Langbehn et al. have introduced a similar technique that instead exploits eye blinks as a means to perform small redirections [37]. As part of their suggestions for how to increase the amount of eye blinks a user makes, they suggested making use of bright lights or virtual objects that move towards the eye of the user. This suggestion could also be seen as a use case for distractors. As a result, if accurate and effective eye tracking hardware ends up becoming more common in the future, we might see distractors become an essential building block of future redirection techniques with this technology.

For this thesis, making use of eye tracking hardware is outside of the scope. There are a few reasons for this. First of all, while VR eye tracking hardware is available for experimentation at campus, the accuracy of the tracking drops during moderately fast head turns. This accuracy issue is problematic when using distractors as they may result in fast head turns. The reason for this accuracy drop in general is that the eye tracking hardware is calibrated with the assumption that

the head-mounted display (HMD) stays static on the user's head. If a user then moves their head so the HMD receives even a tiny displacement in its position, the calibration starts to become inaccurate. This displacement could for example pose a problem for detecting eye saccades, while this would likely be less of an issue for detecting eye blinks.

Furthermore, the addition of eye tracking hardware would introduce an additional variable to an already complex experimental environment. Given the already large scope and complexity of this environment, it would be preferable to limit any additional confounding factors. Employing eye tracking hardware would also require individual calibration per experiment participant. This calibration would result in additional overhead, which could take too much time away from other experiment components.

2.6 External Validity of Estimated Detection Thresholds

Given the number of variables that potentially could affect detection thresholds, it would be difficult to think of these as generalised thresholds. For example, the thresholds that Steinicke et al. initially estimated were thought of as a worst case scenario due to the isolated nature of the study [2]. As such, they should be considered as a baseline while higher redirection is possible due to a variety of variables like engagement or distraction. Despite this, Fuglestad's results have shown that this baseline might be significantly lower than that of Steinicke et al.'s estimations due to the potential effect of better hardware [6]. As such, one might start to wonder what role detection thresholds can fulfil as their generalisability is limited by many factors.

Despite this limitation, there are still a variety of use cases that the estimation of detection thresholds could be applicable for. One of these would be as a tool to measure how much a variable can affect noticeability. By estimating baseline thresholds for a virtual experience and estimating a second group of thresholds whenever a variable is introduced or active, we can make direct comparisons in how the detection of redirection changes. As long as the context of the estimations is taken into account, detection thresholds can also provide a ballpark estimate for the expected amount of unnoticeable redirection. Given that similar contexts are employed by others, they should also receive similar results. If large volumes of detection thresholds were to be estimated across many different contexts and virtual environments, they might also help with providing an overview of the degrees which various variables can affect noticeability.

This discussion is primarily relevant to RQ_1 and the estimation of detection thresholds. In particular, this research makes use of the first example when discussing use cases for detection thresholds where two groups of thresholds are estimated in one experiment. The first group consists of baseline detection thresholds for the virtual experience when no concrete distractors are active, while the second includes detection thresholds for when a concrete distractor is active. This approach allows for direct comparisons on how the introduction of concrete distractors changes the noticeability of redirection. This method of comparison does come with one caveat though. Estimating two groups of detection thresholds in the same amount of time as one would estimate one group results in lower accuracy. Despite this, as detection thresholds in general cannot be taken as entirely accurate due to the number of variables that can potentially affect the result, some error should be

acceptable.

2.6.1 Use of Informed/Uninformed Participants for Redirection

Among the various variables that can affect the noticeability of redirection, awareness of redirection could need some additional discussion. Steinicke et al. mention in their study on detection thresholds that it was possible to increase unnoticeable redirection when the user is unaware of it [2]. This unawareness has since been part of various experiments within redirected walking. For example, all the current state of the art studies on distractors did not inform their participants of redirection before the experiment [19, 30, 31]. While this can provide higher unnoticeable gains, one important question to think of is how applicable this would be in the real world.

If we think of a real-world scenario where a successful application or game makes use of redirected walking, would it be realistic to assume that every single user is unaware of redirection? Since we live in a day and age where the internet keeps us connected, if some users discover the employment of redirection, this information can quickly spread to many others. This spread of knowledge could in turn result in the redirection becoming more easily noticeable. At the same time, Steinicke et al. also mention that being engaged in primary tasks or activities have a similar effect [2]. If a user then knows of the redirection but is engaged enough into the virtual experience, it might help to counteract the otherwise increased noticeability. Of course, it would not be realistic to assume that all users know that redirection is used either as reality would most likely be somewhere in-between.

The challenge lies in trying to design experiments that are closer to what we would expect in the real world. Informing participants to some degree is for example necessary when estimating detection thresholds as they should understand when they should report a detection. If the detection threshold gains are then used in a second experiment to test the performance of redirection, using uninformed participants would not change any results as the gains stay static. As such, for this thesis in particular, it would make the most sense to keep participants informed on redirection for all experiments. This thesis tries to move towards a middle ground between fully informed and uninformed participants. Instead of explicitly mentioning every detail relating to what redirection techniques that are used, participants are mostly given a brief introduction to the topic. They are then asked to notify whenever they detect that the virtual experience feels inconsistent or sluggish. Furthermore, a small amount of deceit is used by mentioning a few additional quality metrics to limit potential priming effects. The information and consent sheet that is given to participants can be found in Appendix D.

An additional benefit of not making use of fully uninformed participants is that the potential sample broadens as some participants could be reused between experiments. As already mentioned in the previous section, it is vital to consider the context of the estimated detection thresholds when looking at the results. To help with this, the context relative to the currently established variables is summarised when presenting the results in Chapter 5.

2.7 Ethics of Estimating Detection Thresholds

One problem with researching redirected walking is the ethics around potentially causing cybersickness for participants. This can be particularly problematic when estimating detection thresholds with standard methods like Steinicke et al. have used [2]. Their method for estimating detection thresholds makes use of a two-alternative forced choice task with a uniform distribution of gains in a given range. These gains are then tested in random order where the participant has to answer whether the current gains are higher or lower than the norm. The problem with this approach is that this range of gains could have values that are high or low enough to result in cybersickness. As such, it could be considered to be unethical to estimate detection thresholds like this. Instead, this thesis makes use of an alternative method which is presented in Section 5.1.2.

Another thing that should be noted is that all participants throughout the experiments in this thesis were told to stop the experiment at once if they experienced any cybersickness or nausea. While prematurely ending the experiment with a participant does not provide entirely accurate data, it can still hold some value for analysis.

2.8 Accessibility of Redirected Walking to Virtual Reality Developers

If a developer is interested in or wants to test redirection techniques, they may be met with a relatively sizeable theoretical bar of entry. This bar of entry is mostly a result of the knowledge around redirected walking primarily being in the academic sphere with very few practical tutorials or code examples for how redirection techniques can be implemented. The possibility of having several key articles locked behind pay-walls does not help with this either.

Despite this, there are some publicly available code repositories that can be looked at. A link to Fuglestad's implementation of various redirected walking elements in the Unreal Engine can for example be found in his masters thesis [6]. Azmandian et al. have developed a redirected walking toolkit for the Unity Engine which consists of implementations for the S2C and S2O algorithms as well as some basic reset techniques [38]. It should be noted that this toolkit has not received any updates or additional features since its release in 2016. The work done with this thesis verifies that the toolkit is working with Unity's 2018.3 version.

The Redirected Walking Toolkit by Azmandian et al. is used as a base for implementation in this thesis. This choice is primarily a result of the author's preexisting familiarity with the Unity Engine, which allowed for a faster development process.

2.9 Choosing a Sufficient Reset Technique

One important thing to consider when developing a virtual reality experience that makes use of redirected walking is the reset technique that will be used. It should not be expected that redirection will be able to fully steer a user away from walls at all times, hence the necessity of these fail-safes. Regardless, there are quite a few different reset techniques available, potentially making it a challenge to choose the right one.

There are three standard reset techniques which were initially presented by Williams et al. [27]:

"Freeze - Backup": The user is notified that they have reached the physical bounds and their virtual position is frozen. They are instructed to take steps backwards until the experience is unfrozen.

"Freeze - Turn": The user is notified that they have reached the physical bounds and the HMD display is frozen. They are instructed to turn 180 degrees, after which the display is unfrozen.

"2:1 - Turn": The user is notified that they have reached the physical bounds. They are instructed to turn 360 degrees virtually while a rotation gain of 2 is applied. This rotation results in a 360-degree turn in the virtual world and 180 in the real world.

Among these three, the 2:1 Turn technique is most commonly seen. This popularity could be due to its relative simplicity and what could be considered as a lower degree of intrusiveness compared to its alternatives. Freeze - Turn and Backup might also be seen as too jarring for users as they directly remove some control in terms of head and body movement in the virtual world.

While the 2:1 Turn technique is the most commonly used one, it does come with a few problems of its own. The high gain value of 2 could for example become uncomfortable for participants even if it only stays active for a short while. Another issue is that it is possible to become stuck between two adjacent sides of a room that are close to the corners. This issue is illustrated in Figure 2 and results in the user moving back and forth with resets happening in perpetual succession.

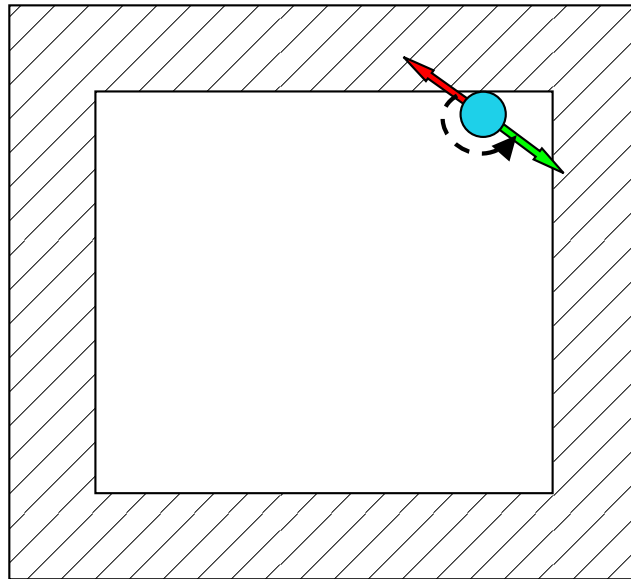


Figure 2: A potential problem that can occur when using the 2:1 Turn resetting technique. The blue circle denotes the user. The red and green arrows indicate the facing direction before and after resetting. The shaded area is the bounds for triggering the reset technique.

To deal with this problem, Nguyen et al. have provided two alternative resetting techniques [39]:

"To Center": This resetting technique tells the user to rotate 360 degrees virtually. A gain value is precomputed so that the facing direction of the user ends up towards the centre of the tracked space once the rotation is finished. As a result, the user rotates 360 degrees + an additional angle to offset them towards the centre of the tracked space.

"To Corner": This reset technique works similarly to the "To Center" technique but instead targets the furthest corner in the room.

These resetting techniques were shown to provide significantly less resets in total over the 2:1 Turn technique in a simulated scenario. Despite this, the authors did not consider an important practical problem that their resetting techniques can result in with current day HMD's. This problem is related to the physical tethering that most modern HMD's still use today. When rotating over 360 degrees in the real world, the tethered cable can easily get tangled and the user will need to avoid wrapping themselves in it. This wrapping is less of an issue for the 2:1 Turn reset as the user only rotates 180 degrees in the real world.

To mitigate the various problems and issues that current day resetting techniques consist of, this thesis has created a variation of the Freeze - Turn reset. This new reset, dubbed as "Pause - Turn - Centre" is detailed in Section 4.6.

2.10 Overall Answers to Research Questions

Since the generated taxonomy has partially operationalised various elements of distractors, it would be prudent to update the terminology of the research questions as well. The updated research questions are as follows:

RQ₁: How noticeable is redirected walking with fully integrated distractors in a playful virtual environment?

RQ₂: Given the highest unnoticeable gains, how effective is redirected walking with fully integrated distractors in a playful virtual environment?

The following two paragraphs summarise how the related work is relevant to the research questions.

RQ₁

For the first research question, a variety of variables that can affect detection thresholds in redirected walking have been mentioned. These have been considered when designing the experiment and corresponding virtual experience. By focusing on an engaging experience designed around fully integrated distractors, it might be harder to notice the redirection, which in turn could allow for stronger gains. A method for measuring detection thresholds has also been provided by Steinicke et al. [2], but it is not used due to problems related to ethics and generally not working well with the theme of fully integrated distractors. The current state of the art is focused on fully integrated distractors and the literature sample provided no research measuring detection thresholds for these.

As such, measuring the detection thresholds for fully integrated distractors should be a reasonable contribution to the field of distractors in redirected walking.

RQ₂

Azmandian et al. have provided a method to measure the effectiveness of redirection by recording the total number of forced reorientations using standard reset techniques [3]. Since standard reset techniques can be included as fail-safes for distractors, this is relevant to make use of for this thesis. Furthermore, this method also allows for comparison with a control condition to find the relative increase in effectiveness.

Breaks in presence and cybersickness are thresholds that can exist at higher gains than detection. Due to this, it would be reasonable to focus on the highest unnoticeable gains as the risk of negative impact on the experience for participants should be minimal.

2.11 Usage of Distractors in Other VR Research Fields

While distractors are just one sub-field of redirected walking, there are similar concepts used in other VR research fields as well. One of the already mentioned ones is using cues in cinematic VR as a means to guide the attention of the user towards areas of interest during watching [29]. Another area where we can see the usage of distractors is in the medical field where they are used for pain relief [40, 41]. In this field, abstract distractors are used as a means to reduce the pain that patients experience under burn injury care or to reduce experimental pain [40]. Furthermore, using VR distractions has shown to significantly decrease the amount of reactive pain behaviour in comparison to traditional distraction methods [41].

In general, we can see that the concept of distracting someone for either their own benefit or for entertainment can be employed in various ways in different fields of VR research.

2.12 Background: Transformation Hierarchies in Game Engines

As part of Chapter 4, there will be more technical implementation details on how the various redirected walking parts of the solution are implemented. In order to understand these technicalities as well as Azmandian et al.'s toolkit [38], it is first necessary to provide some background on one important topic: Transformation hierarchies in game engines.

A transformation hierarchy in computer games is a relatively simple concept, but can result in rather complex behaviour. This hierarchy is a specific subset of functionality from a concept known as scene graphs, which used to logically couple together objects in a 3D scene. The gist of a transformation hierarchy is that objects or entities can be parented to each other. When parented, the child will no longer be positioned directly in world space, but rather relative to its parent. As such, if a parent moves, so will all of its corresponding children who may have individual relative positions within the hierarchy. This cascading effect does not only have to be in respect to position changes, but also scale and rotation. If we for example have a car object in a game with all the individual parts as children of a root node, then we can modify the position, rotation or scale of the root and it will propagate to each of the children. This way, we can rotate, re-position or scale the

entire car by merely modifying the root node in the hierarchy.

2.13 Background: The Redirected Walking Toolkit

Since the implemented solution for this thesis makes use of the redirected walking toolkit [38] as a base, it is first necessary to at least have an overview of it.

2.13.1 How Redirected Walking is Implemented in the Toolkit

An interesting side effect of transformation hierarchies is the perception of change for the child nodes. This effect is essential to how redirected walking is implemented in the redirected walking toolkit. To best illustrate this effect, it is easiest to provide an example. Consider that we have a parent: **"Redirection Root"** with two children **"Camera"** and **"Physical Space Representation"**. The camera can move around in the representation of physical space and both children will always be correctly mapped with each other, regardless of how the parent's transformation changes. As children, from their perceived point of view, they are consistent with each other due to how transformation hierarchies function. The interesting part becomes if we place our virtual world outside of hierarchy that the root governs, or rather have root and virtual world on the same hierarchical level.

If we now rotate the redirection root, the children will have the perception that it is the virtual world itself that rotates, and not themselves. Despite what the children perceive, it is their redirection root parent that rotates and not the virtual world itself. This effect is the essence of how redirected walking is implemented in this case. Both curvature and rotation gains are implemented by injecting small amounts of rotation to the y-axis of the redirection root as the child camera moves and looks around. By doing so, it is possible to create the perceived effect that the child camera rotates more or less in the virtual world, compared to the real one. In a similar vein, it is possible to create curved paths.

2.13.2 Toolkit Terminology

The toolkit uses slightly different semantics and terminology compared to what we could consider as the standard set by Steinicke et al. [2]. For the sake of consistency with the toolkit itself, this terminology will be used forwards in the thesis. The terminology that is different and their equivalents are as follows:

Rotation Gains

The toolkit does not semantically think of rotation gains as a multiplier to head movement, but rather as a percentage increase or decrease which are respectively named positive and negative rotation gains. For example: a rotation gain multiplier of 2, meaning that head rotations are twice as fast in the virtual world would be the equivalent to a positive rotation gain of 1. In this case, it is a 100% increase over the base rotation. A multiplier of 0 would respectively be equivalent to a negative rotation gain of -1, meaning that the head no longer can rotate on the y-axis.

Curvature Gains

Instead of defining curvature gains as a ratio relative to the radius of the circular arc, the redirected walking toolkit defines curvature gains as curvature radius. In this case, a curvature radius of 5m means that the user's curved path will be on the edge of a circle that has a radius of 5m.

Forced Reorientation Techniques

"Forced reorientation techniques"/"ROTs" are generally either called by this semantic name or simply as "resets". There is no direct consistency in the literature regarding which one researchers use, but the toolkit uses "resets" as the term for forced reorientation techniques.

2.13.3 Toolkit Structure

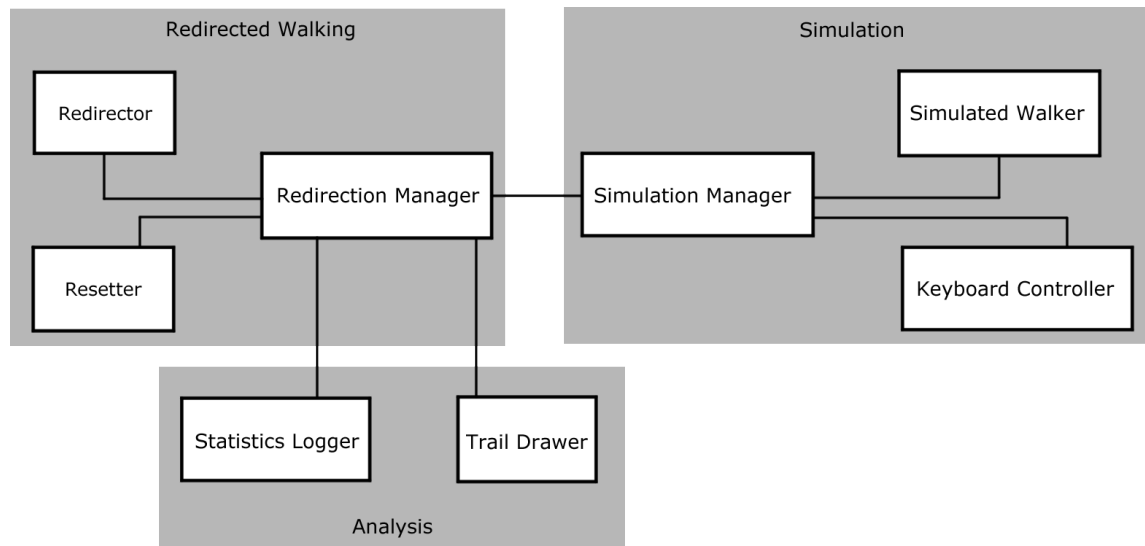


Figure 3: This illustration shows a recreation of Azmandian et al.'s figure [38] which gives an overview on the structure of the redirected walking toolkit. The structure has been further extended by the work in this thesis as discussed in Section 4.2. The extended structure can be seen in Figure 4.

The redirected walking toolkit's overall structure is divided into three parts. These can be seen in Figure 3 and consist of redirected walking, simulation and analysis components. The work in this thesis is primarily related to the redirected walking side of the toolkit and as such, this part will be further detailed.

RedirectionManager

The main control point of the toolkit is the `RedirectionManager` script. The script itself is attached to the root object in the redirected walking object hierarchy and as the name implies, manages the whole solution. This management includes keeping track of the strength of gains, calling virtual functions on redirectors and resetters when applicable, and facilitating general communication

between all of the components in the toolkit.

Redirectors

Redirectors are the scripts in the toolkit that manage the injection of camera angles which results in redirection. All redirectors have to extend a base `Redirector` script which makes use of various virtual functions that are called as necessary by the `RedirectionManager`. The `S2C` and `S2O` redirectors are examples of scripts that in this case extend the base redirector class. This class allows for a generic approach where any type of redirector could be developed as long as it complies with the structure and virtual functions of the base class.

Resetters

Resetters are similarly structured to redirectors. A resetter is capable of the same functionality as a redirector, but is only active whenever the user leaves a defined safe area within the physical space. By default, a resetter is triggered whenever the user is 0.5m away from the edge of the physical space. Furthermore, a resetter has additional callbacks for when a reset has triggered and when it is finished to allow developers to easily work with these events.

This concludes the background needed to understand the implementation details of Chapter 4. Before moving on to this chapter though, the general methods that were applied in this thesis are disclosed in the following chapter.

3 General Methods

This chapter consists of methods that are relevant to the thesis as a whole. Specific methods used for each of the two experiments can be found in Chapter 5 and Chapter 6.

3.1 Literature Acquisition

In order to find relevant literature for the literature analysis in Chapter 2, a variety of search terms and databases were used. The list of keywords, keyword combinations and literature databases that were used can be found in Table 3. Similar searches were also conducted with the ACM digital library database, but these searches did not provide any additional literature that had not already been found in other searches. As such, these queries are not included in the table. As a side note, these queries are a refreshed and combined version of previous literature searches in the IMT4205 - Research Project Planning and IMT4894 - Advanced Project Work courses. The following paragraphs detail how the chosen literature was acquired and the employed filtering methods.

General Literature on Redirected Walking and Detection Thresholds

For the first three queries in Table 3, literature was picked for reading as long as the title or abstract focused on either detection thresholds or user comfort. The focus of the queries was to find recent development within the field of redirected walking and the estimation of detection thresholds. All result pages were scanned through in these three queries.

Distractors and Attention

Table 3 also consists of three queries related to distractors. The first of these was conducted to look for state of the art applications of distractors while the second and third were used to acquire background literature on the topic. For the first search, all pages of query results were scanned through. For the second and third searches, the first ten pages of results were scanned through due to the higher sample.

One last query related to cinematic VR and attention was added to look for related work in this field. The ten first pages of results were scanned through for this query. Literature was chosen for reading based on similar criteria as the queries in the previous section.

Literature Acquired Through Citations

As a secondary approach to acquiring literature, an additional 9 papers were found through citations in papers that were chosen in Table 3.

Literature Acquired Through Discussions With Supervisors

3 additional research papers were acquired through discussions with the supervisors. These consisted of various research methodology papers as well as some pointers towards medical research that makes use of distractors in VR.

Keywords	Database	Combination	Filter	Results	Chosen for Reading
"Redirected Walking" "Threshold" "Thresholds" "Comfort"	Google Scholar	("Redirected Walking") AND ("Threshold" OR "Thresholds") AND "Comfort"	After 2014	69	7
"Redirected Walking" "Threshold" "Thresholds"	Google Scholar	("Redirected walking") AND ("Threshold" OR "Thresholds")	After 2018	119	6
"Redirected Walking" "Threshold" "Thresholds"	IEEEExplore	"Redirected walking" AND ("Threshold" OR "Thresholds")	None	13	5
"Redirected Walking" "Distractor" "Distractors"	Google Scholar	("Redirected walking") AND ("Distractor" OR "Distractors")	After 2016	76	3
"Redirected Walking" "Distractor" "Distractors"	Google Scholar	("Redirected walking") AND ("Distractor" OR "Distractors")	None	172	3
"Virtual Reality" "Distractors" "Redirection"	Google Scholar	("Virtual Reality" AND "Distractors" AND "Redirection")	None	186	3
"Cinematic VR" "Attention"	Google Scholar	("Cinematic VR" AND "Attention")	None	123	2

Table 3: List over keywords and combinations that were used for the literature search.

This puts the total amount of literature that was acquired at **41** papers for the literature review in Chapter 2.

3.2 Personal Data Collection and GDPR Compliance

Throughout the experiments, a demographic questionnaire was used to gather certain pieces of personal data. This section is dedicated to justify why each piece of personal data was collected and provide validation that the study itself is GDPR compliant.

3.2.1 Demographic Questionnaire

The demographic questionnaire that was employed in both experiments can be found in Appendix C. It consisted of various demographical questions, as well as optional qualitative feedback questions.

The reasoning for using the following demographical questions is as follows:

Gender: Given that there is prior research on the potential effects of gender on sensitivity to redirection [20], this is a relevant piece of data to collect.

Age Range: There may be cognitive differences between younger and older participants that can affect the result. Age ranges are used instead of direct age so that recognising individuals is more difficult.

Whether the participant needed to remove any optical corrections while in VR: If a participant needs to remove any optical corrections to make use of a VR HMD, the sharpness of their vision may or may not be compromised. This loss of sharpness could in turn have some effects on their sensitivity to redirection as it for example could affect the perceived optical flow.

Whether the participant has taken part in prior redirected walking experiments: This question was asked in case there are any trainable effects towards redirection. This could be the case as there may be similar effects to the possibility of mitigating cybersickness through training [25]. Furthermore, research suggests that adaptation effects exist for curvature gains [20].

How much prior VR experience the participants considers they have: In a similar vein to the previous question, prior experience with VR could have some effect on how comfortable participants are in VR. This comfort level could in turn have some effects on how comfortable they are with using redirected walking.

One question that might have been relevant would be how much prior game experience participants have had. This question was not included in the questionnaire as research has suggested that this does not factor into the noticeability of redirection [20].

3.2.2 GDPR Compliance and Data Anonymity

In order to validate that this study was GDPR compliant and following other relevant regulations, an application form was sent to the Norwegian Centre for Research Data¹ (NSD). Their response, showing that the study is compliant with GDPR as well as their own terms, can be seen in Appendix E. The following two paragraphs consist of a summary of the information that was provided in the application to NSD.

The demographic data that was gathered made use of a paper questionnaire and required written consent. These two were tied together with a numerical ID which could be used if necessary to remove any personal data if requested. Participants were asked at the end of each experiment if they wished to be given their ID so it could be used for GDPR related requests. The demographic data and written consent were stored in a securely locked box at campus until data processing was necessary. Once processing was necessary, the paper data was transcribed and stored on the author's private home directory at NTNU's servers². This data was then processed using NTNU's software

¹<https://nsd.no/nsd/english/index.html>

²<https://innsida.ntnu.no/web/guest/wiki/-/wiki/English/Your+files+on+the+NTNU+server>

farm³. The demographical data on the private home directory as well as the paper information are planned to be destroyed on the thesis hand-in date.

Outside of the demographic information that was recorded, each of the two experiments recorded a variety of software side performance data. This data was also tied to the demographic information with a numerical ID, but stored in the project repository on GitHub [42]. This data is openly available as there is no way to tie it to any individuals without access to the demographic information.

3.3 Development Environment

The development environment that was used to develop the Ensemble Retriever VR game consists of:

- Unity Engine, version 2018.3.5f1.
- Microsoft Visual Studio 2017 Integrated Development Environment.
- SteamVR Unity plugin for virtual reality development.
- A Mersenne Twister library [43] for cases where high-quality randomness is necessary.
- Azmandian et al.'s Redirected Walking Toolkit [38] for providing base redirected walking functionality.
- OpenVR Advanced Settings⁴ to disable SteamVR's internal chaperone system.

Further details on Ensemble Retriever and additions to the redirected walking solution can be seen in Chapter 4.

3.4 Hardware Environment for Experiments

Throughout the two experiments, a desktop computer with the following specifications was used:

CPU: Intel i7-6700k

GPU: Nvidia Geforce GTX 1080

RAM: 16 GB

Operating System: Windows 10 Pro

Together with this desktop computer, an HTC Vive HMD + Vive Controllers was employed together with 5m cable extensions to allow for full traversal of the physical tracking space without any significant tethering issues. The physical space that was used can be seen in Section 5.1.5. Finally, a pair of Audio Technica ATH-MSR7BK headphones were used to provide sound for participants.

3.5 Software Environment for Experiments

On the software side, the developed Ensemble Retriever game was used while running in the Unity Editor. The reasoning for using the software in the editor rather than as a built version was that the

³<https://innsida.ntnu.no/wiki/-/wiki/English/software+farm>

⁴<https://github.com/OpenVR-Advanced-Settings/OpenVR-AdvancedSettings>

employed version of Unity had build times that took several hours. This built approach would be rather inflexible in case any apparent issues appear during experiments that require rapid fixing. Instead, a more flexible option was used by running the game in the Unity Editor, allowing for small changes or fixes to be quickly implemented if needed. This approach does come with a small trade-off in terms of ingame performance as the editor results in higher overhead. This overhead was considered as an acceptable trade since the frame rate of the game mostly stayed at 90 frames per second. SteamVR's motion smoothing functionality⁵ was used for situations the target frame rate of 90 could not be held.

The implementation specific details of Ensemble Retriever is discussed in the following chapter. This background is essential before moving on to the two experiments which have been conducted.

⁵<https://steamcommunity.com/games/250820/announcements/detail/1705071932992003492>

4 Implementation

This chapter consists of implementation details for all relevant redirected walking components as well as an overview of the game design for the developed "Ensemble Retriever" game. The game design overview is used to provide examples and documentation on how distractors have been fully integrated into the experience itself.

4.1 Open Source Repository - GitHub

The source code and project assets for Ensemble Retriever can be found in a publicly available GitHub Repository [42]. It should be noted that the game itself is an extension of a small prototype that was previously developed for the IMT4894 - Advanced Project Work course. This prototype consisted of no redirected walking elements and featured a battle in VR against an angry contrabass enemy. The majority of the source code has needed to be rewritten or refactored to facilitate a more generic architecture that supports the larger scope of the current game.

4.1.1 Licensing and Attribution

Ensemble Retriever makes liberal use of royalty-free assets as a means to fasten then development time of the game. As such, it is also necessary to properly provide attribution to these assets. In general, each royalty free asset in the repository includes a corresponding license file which details the specific license that applies to it. The following list gives an overview of the royalty-free assets that were used for the project:

- Most Particle effects.
- Fonts.
- Some 3D Models like hats/crowns, conducting baton, objective arrow and the cave walls in the "Hall of The Mountain King".
- Skybox.
- Music.
- Sound effects.

Those with interest in reusing or extending the project need to follow the licensing terms that apply for all of these assets. There are also specific 3D models that were reused from a previous project with permission from their creator, Yijie Zhou. These can be found in the "*Assets/Meshes/BlenderAssets/*" folder of the repository and include:

- All instrument 3D models.
- The virtual environment, excluding the "Hall of the Mountain King".

Extension of these assets are not permitted in other work and need to be replaced as the permission

was given to reuse these assets for this project specifically. If the project is to be used for the sake of reproducible results, then these assets do not need to be replaced. All other assets were developed specifically for the Ensemble Retriever game and include:

- All game logic + extensions to the Redirected Walking Toolkit.
- All animations.
- All sprites/textures except for an image of an HTC Vive Controller.
- All voice acting.
- Some particle effects like player attacks, projectile blocks and sweating particles for The Mountain King.

These elements of the game are under a MIT License and allow for reuse/extension as long as the license terms are held.

4.2 Redirected Walking Toolkit - Extended Code Architecture

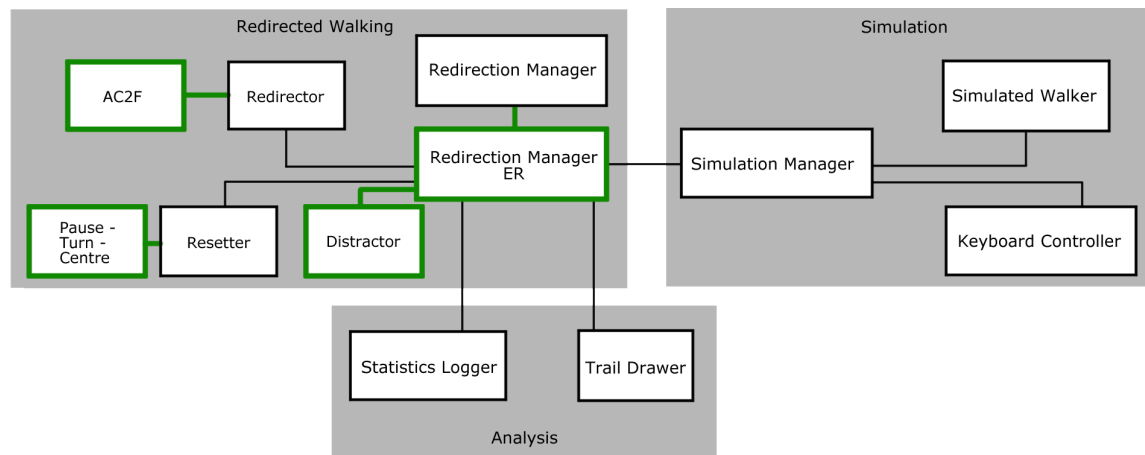


Figure 4: This illustration provides an overview on how the structure of the redirected walking toolkit has been extended. New additions are marked with the use of green outlines. The original structure can be seen in Figure 3.

As part of developing Ensemble Retriever, Azmandian et al.'s Redirected Walking Toolkit [38] has been extended to support the usage of distractors and any other interfacing that the game has required. A chart showing the general additions to the toolkit's architecture can be found in Figure 4. The general thought process throughout the development of these extensions was to not do any significant changes to the toolkit itself for the sake of keeping its modular structure. If functionality could be built on top of parts of the toolkit, then inheritance was used. If any part of the toolkit required major changes, it would be copied to a separate file which these modifications were written in. As far as minor changes are concerned, these mostly consisted of changing some data access properties of variables to allow for easier communication between classes or bug fixes.

To supplement the general overview in Figure 4, the following list provides slightly more details on the exact extensions that have been implemented:

- `RedirectionManagerER` which extends `RedirectionManager` to facilitate communication and management between new components.
- The "Align Centre To Future" (AC2F) Redirector. This redirector is aimed to be used while standing still.
- The "Pause - Turn - Centre" Resetter.
- The Distractor Trigger System.
- A `ExperimentDataManager` script which handles data collection and experiment management.
- A `GainIncrementer` script which is used as part of Experiment 1.

Each of these will be further detailed in the following sections.

4.3 Managing The Extended Architecture

The `RedirectionManagerER` script [44] works in a similar way to the base class it extends. It facilitates communication between all of the new distractor related components and a few other things. These include switching between S2C and AC2F whenever distractors trigger, sampling position changes to calculate future walking directions, checking for alignment with future path during distractor battles and some additional debug related functionality. In this case, the manager makes use of two redirectors which consists of the S2C and AC2F redirectors. S2C is used when walking due to its rather generic nature, while AC2F is used during distractor battles as it is more specialised in terms of how it redirects.

Distractors in general function very similarly to resetters. They have associated callbacks and triggers which can generically be extended for whatever use the developer needs. They also have their own trigger safe bounds, which in general should happen somewhat earlier than what a reset would. It is necessary to keep in mind that once a distractor triggers, the user will need time to be able and stop walking without hitting the reset trigger bounds. The size of this buffer will mostly depend on the walking speed of users, but it is useful to try and find a balance in terms of the distance between these two bounds. The reasoning for this is that a large movable space before triggering any distractors would be preferable as they might trigger too often otherwise. This buffer was originally 0.5m, but has since been increased to 1m due to observations during Experiment 1.

4.4 Distractor Enemies

There is no directly generic script or class for distractors in the provided solution as there are very few strict guidelines on how these can be made. In general, `RedirectionManagerER` stores a generic list of all potential distractor objects that can be spawned. Whenever the user reaches the distractor trigger bounds, the `RedirectionManagerER` script will in this case spawn a distractor from its list.

A distractor's primary responsibility after being spawned is to notify the manager whenever it is finished so that the manager can call relevant callbacks and clean up after the distractor.

Other than this, any behaviour could be programmed into the developed distractor. For Ensemble Retriever, all distractors extend a `DistractorEnemy` script [45] which handles this responsibility in conjunction with some generic enemy behaviour. In a more general solution though, some generic base distractor class would likely be present.

4.5 The "Align Centre to Future" Redirector

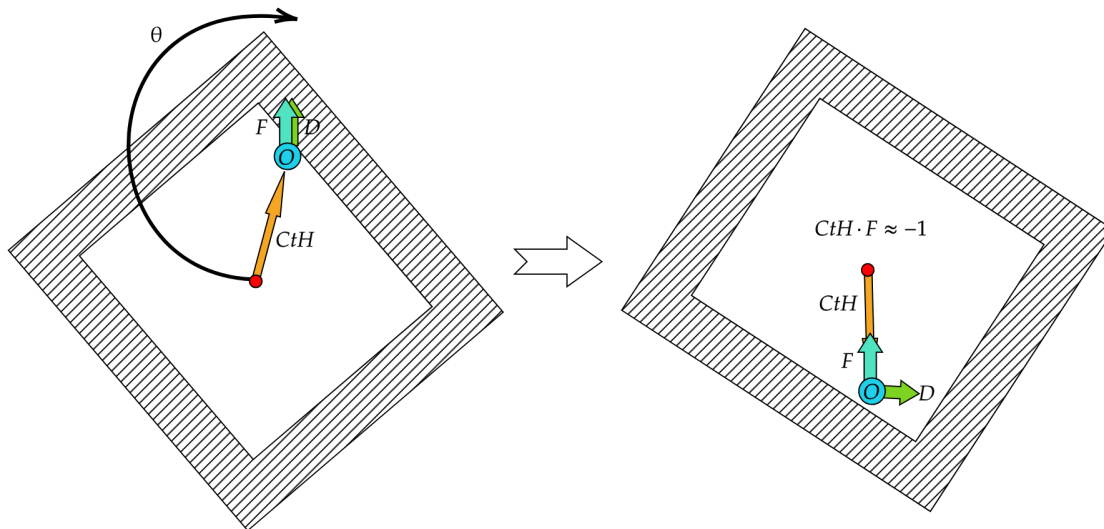


Figure 5: AC2F aims to choose the rotation gains that bring the physical centre to head (CtH) vector in alignment with the future virtual walking direction (F) of the user. The user's current facing direction (D) is not used other than setting the value of F as the algorithm starts. The origin of this reorientation is at the position of the user (O).

The "Align Centre to Future" (shortened to AC2F) redirector is based on what Peck et al. [11] as well as Chen and Fuchs [30, 31] have mentioned in terms of a modified future path driven S2C algorithm. Neither of their research provided any source code for how they modified S2C and as such, AC2F is loosely based on Azmandian et al.'s S2C implementation, which is part of the redirected walking toolkit. AC2F relies on rotation gains and an alignment heuristic to align the user's future virtual path towards. The definition of the user's future virtual path in this case consists of a heuristic that samples the last second of their positional changes. The alignment heuristic on the other hand is the centre of the physical space. An illustrated view on how the algorithm works can be seen in Figure 5. The algorithm aims to choose between negative or positive rotation gains in a way that results in closer alignment towards the heuristic. The source code for the algorithm can be found in the project repository [46].

4.5.1 Algorithm Pseudocode

The AC2F algorithm is summarised in the following simplified pseudocode:

```

chosenInjection = 0;
// Check if the change in rotation exceeds the threshold for applying gains
if (deltaHeadRotationAngle >= rotationThreshold) then
    negativeRotationGainInjection = deltaHeadRotationAngle * negativeRotationGain;
    positiveRotationGainInjection = deltaHeadRotationAngle * positiveRotationGain;
    dotProductFromNegativeInjection = Dot(negativeRotationGainInjection * physicalCentreTo-
Head, futureVirtualWalkingDirection);
    dotProductFromPositiveInjection = Dot(positiveRotationGainInjection * physicalCentreToHead,
futureVirtualWalkingDirection);
    if (dotProductFromNegativeInjection < dotProductFromPositiveInjection) then
        chosenInjection = negativeRotationGainInjection;
    else
        chosenInjection = positiveRotationGainInjection;
    end if
end if
redirectionRoot.Inject(chosenInjection);

```

The gist of the algorithm is that it checks which of the two types of rotation gains that result in the best alignment towards the future virtual walking direction. In this case, dot products are compared with an alignment goal of ~ -1 . This value is the dot product for when the user's future virtual path points in the opposite direction of a vector between the centre of the room and the user's head. Once a gain has been decided for use, it is injected to the root of the redirected walking hierarchy. The angle of this injection consists of the change in head rotation between two frames multiplied by the chosen gain.

Given that these calculations run every frame, it may not necessarily be the best solution in terms of performance. Despite this, the implementation was chosen to work this way for the sake of readability and simplicity as the performance cost was negligible.

Hypothetical Optimisation of AC2F

In the hypothetical case of it being necessary to optimise AC2F, there are a few useful technicalities that could be considered. One of these is that the algorithm always will rotate the redirected walking root in either a clockwise or counterclockwise fashion depending on which of these takes the shortest time. Head rotations will also either be in a clockwise or counterclockwise direction. Applying a positive rotation gain to a head rotation will result in the redirected walking root to rotate in the same clockwise/counterclockwise direction. This effect means that the root will rotate slightly with the head rotation. Applying a negative rotation gain on the other hand will result in the opposite clock direction, meaning that the root will rotate slightly against head rotation.

Given this behaviour, it is possible to skip dot product comparisons once the algorithm has decided whether to align in a clockwise or counterclockwise manner. In this case, there will always be a correct gain type to choose depending on which clock direction the head is rotating in.

4.5.2 Smoothing

One problem with rotation gains is that switching between positive and negative gains with large differences makes it very easy to notice redirection. This problem can be mitigated with the use of smoothing components which can be added to redirection algorithms. Particularly, smoothing can be applied by interpolating from a positive gain injection to a negative whenever this change is needed and vice versa. This interpolation creates a more subdued and gradual change rather than quickly jumping from one type of gain to the other. The main challenge with smoothing rotation is that we cannot interpolate the actual gains themselves as these have to stay static. Instead, we need to interpolate the camera injections that happen which will vary frame by frame depending on how much the user's head moves. As such, standard interpolation which has a set start and stop value is not possible as the stop/target value to interpolate towards will change every frame. Instead, the implementation AC2F uses was found on the Unity Forums [47] which is smoothing solved using a differential equation. The formula for this interpolation with the corresponding context is as follows:

$$f = \frac{followerOldValue - targetOldValue + (targetNewValue - targetOldValue)}{(interpolationSpeed * t)}$$

$$followerNewValue = \frac{targetNewValue - (targetNewValue - targetOldValue)}{(interpolationSpeed * t) + f * Exp(-interpolationSpeed * t)}$$

followerOldValue: The interpolated value this formula output during the prior frame.

followerNewValue: The new interpolated result which this formula outputs during the current frame.

targetOldValue: The value of the target during the prior frame.

targetNewValue: The value of the target during the current frame.

interpolationSpeed: The speed of interpolation.

t: Input interpolation time. This value is within the range of [0, 1].

f: An intermediary variable to decompose the equation.

Using this formula allows for smooth interpolation between rotation gain types even though the target may change slightly every frame.

Dealing With Edge Cases

The AC2F algorithm makes use of a threshold for head rotations before applying any gains. The reason for this threshold is to avoid applying gains when the head is relatively still, but still moving due to small head vibrations and tracking inaccuracies. This threshold poses somewhat of a challenge in terms of how to deal with smoothing. If smoothing already is in progress and a head rotation goes below the threshold, then what should the algorithm do? One option would be to disable smoothing when this happens, but this creates an issue with a specific edge case. Whenever a user rotates with their body and head, the stopping motion results in a small bob towards the opposite direction. This small bob is below the threshold for applying gains and results in a somewhat jarring difference between applying gains and not applying them if we disable smoothing.

To deal with this edge case, AC2F allows the smoothing component to continue until it is finished. This approach deals with this specific edge case, but still has some problems of its own which could be improved in the future. If a user moves with only their head instead of head and body, then the smoothing will result in a somewhat sliding rotation effect when the user stops their head. While this effect only lasts for less than half a second, it could still be considered as annoying or unwanted. Fixing this problem while still also dealing with the prior mentioned edge case would be ideal in terms of smoothing.

4.6 The "Pause - Turn - Centre" Resetter

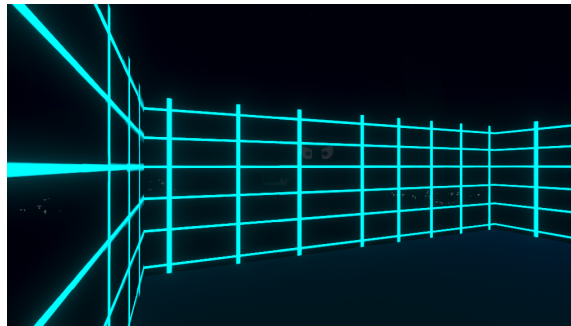


Figure 6: This screenshot shows the Pause - Turn - Centre resetter in action. The virtual world has been mostly faded away and paused while the user can normally look around in the representation of their physical space.

As mentioned in Section 2.9, there are a variety of issues with current resetting techniques which limit their usefulness. To deal with these, a new resetter has been developed for this thesis which is called "Pause - Turn - Centre". This resetter takes inspiration from the Freeze - Turn resetter [27] as well as Sra et al.'s discussions on hiding noticeability through visibility limitation methods [19]. The end result is somewhat different from Freeze - Turn with changes that aim to improve effectiveness and user experience. The source code for the resetter can be found in the project repository [48].

Whenever the Pause - Turn - Centre resetter triggers, it will quickly fade in a chaperone style boundary which represents the physical space. At the same time, a slightly transparent black layer

will be used to fade out the virtual environment so it is only slightly visible. While this resetter is active, a negative rotation gain of -1 is used to effectively freeze the y-rotation of the user in the virtual world. Meanwhile, the user can still look around in the representation of the physical space with normal head rotation. As the name implies, components of Ensemble Retriever that exist within the virtual world are paused until the resetter has finished. The resetter finishes when the user looks towards the centre of the physical space, meaning their walking direction will be towards its centre. A screenshot showing the resetter after being triggered can be seen in Figure 6.

The resulting behaviour avoids the issues that other resetters have in terms of cybersickness and sub-optimal reorientation. Cybersickness is minimised by obscuring the virtual world with a slightly transparent layer as vision limiting techniques decrease noticeability of redirection [19]. It also allows the user to clearly see where the bounds of the physical space are. By instructing the user to look and move towards the centre of the room, there are no possibilities of becoming "stuck" between resets like when using 180-degree turns. The overall rotation that the user has to make is also smaller than 360 degrees, meaning that they will not be wrapped around any physical tethered cables which could cause issues. The trade-off for these improvements is the potential for this to be considered as more disruptive, but it could be argued that the improvements to safety, effectiveness and user comfort are worth this disruption.

4.6.1 Clipping Related Problems

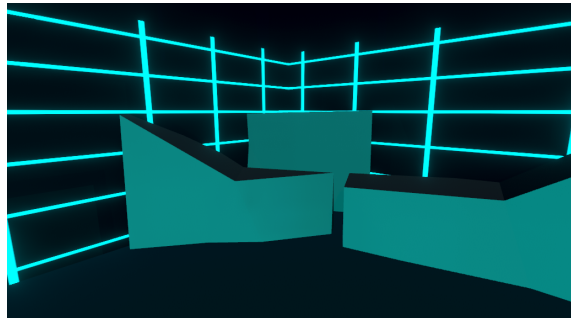


Figure 7: This screenshot shows a prior bug that existed when using the Pause - Turn - Centre resetter. Virtual world geometry has in this case clipped into the representation of the physical space, resulting in rather confusing visuals as the virtual world is effectively paused during this time.

One issue that cropped up during the implementation of Pause - Turn - Centre can be seen in Figure 7. If the representation of the physical space overlaps with other geometry from the virtual world, then this would cause some rather confusing visuals. This confusion stems from the issue that a negative rotation gain of -1 is fully visible for this overlapping geometry. This problem has been solved in the following manner.

Two separate cameras are employed. One of these renders the virtual world while the other renders everything that exists under the redirected walking root and could be considered as the

physical space. By forcing the depth level of the physical space camera to always be closest, we can draw the physical space representation over everything else that exists in the virtual world. As such, even if geometry overlaps into the physical space, this overlap cannot be seen as the physical space representation will be drawn on top of it. Drawing the physical space representation over the virtual world has a benefit and a drawback. It is simple to implement as forcing one camera to render on top of another is as simple as setting a depth value for the camera itself. The downside is that using multiple cameras in VR is rather performance costly as each camera must render twice for stereoscopic vision. As such, there are effectively four render cameras in use instead of two. This approach becomes a more significant performance problem with post-processing as this needs to apply to each camera. Despite this overhead increase, there is some granularity as the post-processing effects that are used can be tweaked for each individual camera. By doing so, it is possible to optimise away specific post-processing effects from the rendering of the physical space representation as it will not be seen very often.

4.6.2 Pausing the Game Using "Pause - Turn - Centre"

In order to properly pause relevant game objects in Ensemble Retriever with "Pause - Turn - Centre", it is necessary to define what is and is not pausable. This definition is solved by having all pausable dynamic objects extend a `Pausable` class. By doing so, each dynamic object acquires access to some callbacks and pause state information. Pausing all pausables is handled by the `RedirectionManagerER` script where it generates a list of all `Pausable` objects and triggers a related pause callback for these. It is then up to each individual `Pausable` object to control their paused behaviour. This approach allows for selective pausing of virtual world objects while allowing elements like the physical space representation and HMD tracking to still function.

4.7 Experiment Management

The experiments that have been conducted for this thesis are managed through two primary scripts: `ExperimentDataManager` [49] and `GainIncrementer` [50]. The `ExperimentDataManager` script takes care of all related data recording for Experiment 1 and 2 while the `GainIncrementer` script is used during Experiment 1 to incrementally and randomly increase gains. Further details on what and how data was recorded can be seen in Chapter 5 and 6 for Experiment 1 and 2 respectively.

4.8 Supporting the y-axis in the Redirected Walking Toolkit

As part of triggering relevant resets and other colliders, the redirected walking toolkit makes use of a "head follower" collider [51] which represents the collider for the user's body. It will continuously be below the user's head position as the user moves around. One issue that was uncovered during development was that this functionality stopped working as intended if the redirected walking root changed y position. The reason for this was that the "head follower" collider did not take these changes into account and continuously attempted to place itself at a y-value of 0 in world space. This approach was not particularly flexible and as such, this functionality of the toolkit has been updated to support the y-axis. This improvement is handled by having the collider place itself at a

local position of 0 rather than in world space, meaning that it will always be below the user due to the transformation hierarchy it exists in.

4.9 Game Design Overview of Ensemble Retriever

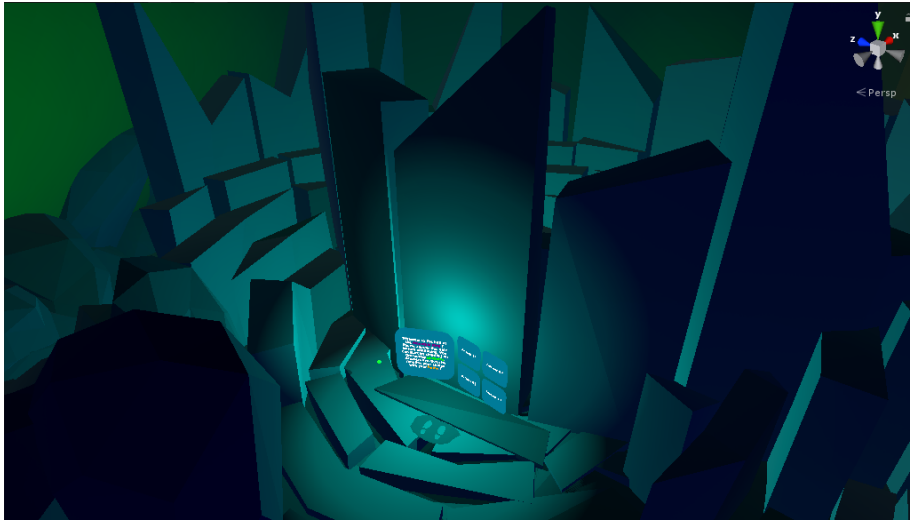


Figure 8: This screenshot from a birds eye view shows the "Hall of The Mountain King" with the quiz section that the player has to answer.

In Ensemble Retriever, the player takes the role of a conductor whose goal is to retrieve the "Mountain King", which is an instrument that has disappeared from their ensemble. In order to find the "Mountain King", the player has to walk around in a large virtual environment and ask the local residents for clues before they can enter the "Hall of The Mountain King" seen in Figure 8 and 9. Throughout the player's exploration of the virtual world, they are attacked by various distractor enemies which have to be defeated to gain experience points and progress the game. These battles consist of having the player use their contrabass shield to absorb enemy projectiles. Once enough projectiles have been absorbed, the player can counterattack with their magical conductor baton. Defeating distractors allows the player to level up over time and choose whether to upgrade their shield size or baton damage. The game culminates with a battle against the "Mountain King" that tests the player's skill and abilities which they have trained throughout their battles with distractor enemies.

The following sections provide a more detailed context on the game itself and how distractors have been fully integrated into the experience.

4.9.1 Virtual Environment

The virtual environment that players walked through can be seen in Figure 10. A top-down view is also available in Figure 11. If taking a straight line from the start to the end, players would need

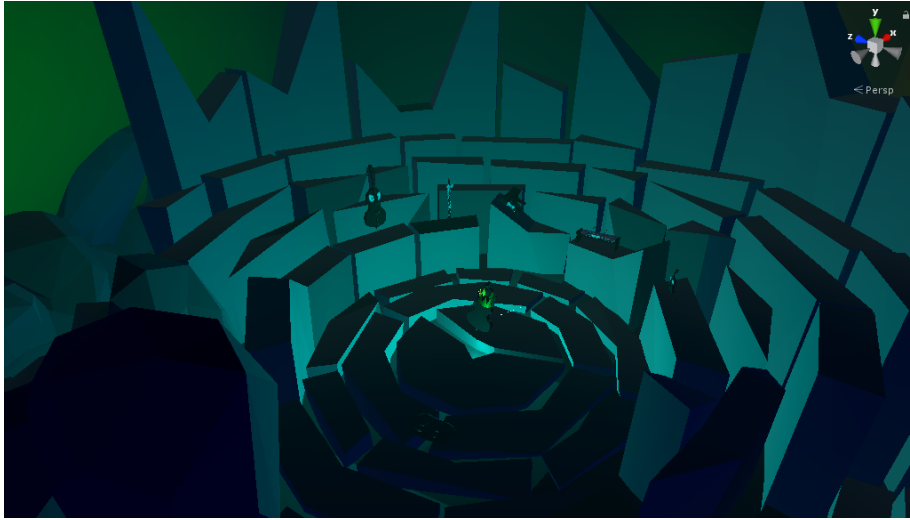


Figure 9: This screenshot from a birds eye view shows the "Hall of The Mountain King" after the player has answered the quiz section in the game. This is where the Mountain King is fought.

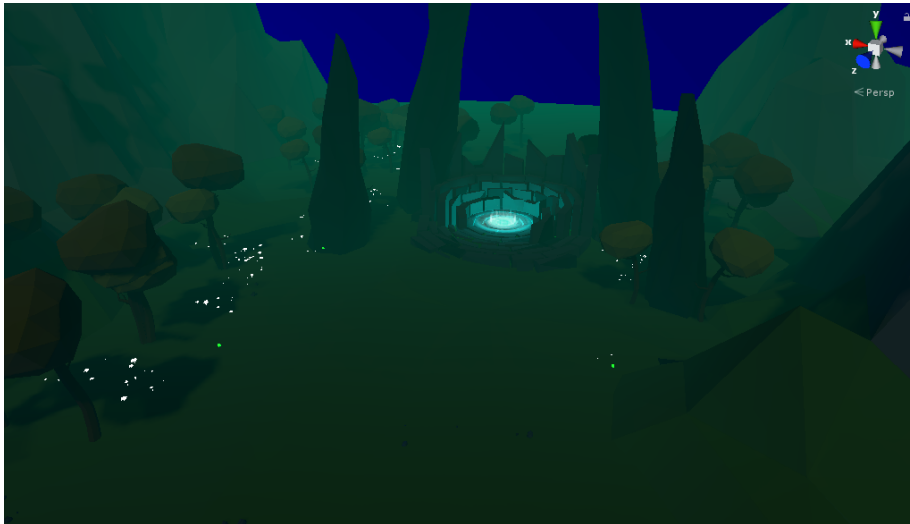


Figure 10: This screenshot from a birds eye view shows the virtual environment of Ensemble Retriever.

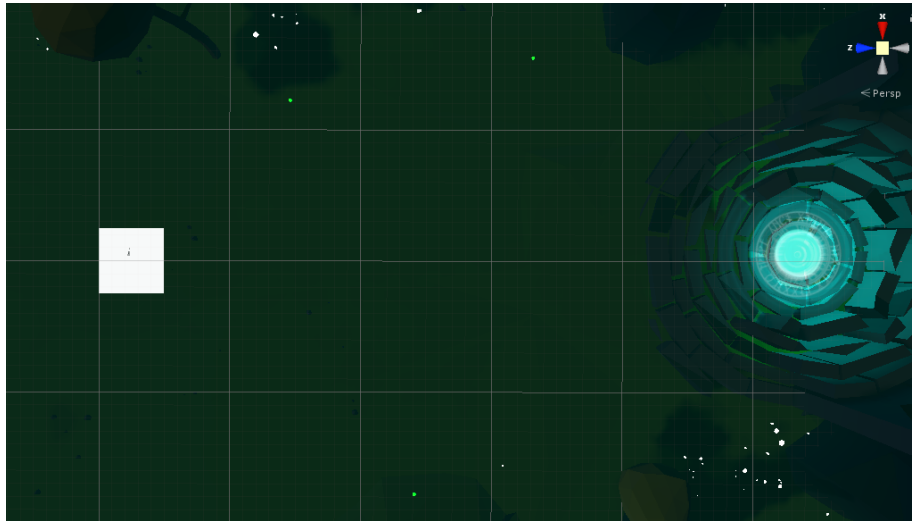


Figure 11: This top-down screenshot of the environment shows the walkable area that players walked through in Experiment 1. In this case, players started where the white square representing the physical space is and moved to the portal on the right. On their path, they had to visit three small green fireflies. It should be noted that each small square in the grid is equivalent to 1m squares in the real world.

to walk $\sim 55\text{m}$, although the actual distance is a bit longer as visiting the three fireflies also was necessary.

4.9.2 Game Flow

This section provides an outline of the flow of Ensemble Retriever and how the game progresses throughout a play session.

Ensemble Retriever starts with a tutorial that teaches the player the basics of the game. As seen in Figure 12, the tutorial is provided to the player by one of the in-game characters. It provides information on the context/story, the player's goal, how to deal with resets, how to fight enemies and what to do if they are lost. In Experiment 1, the tutorial also provides information on what the player should do if they notice they are being redirected. The entire tutorial is voiced in case the text is not sufficiently readable.

After the tutorial is finished, the game transitions to a walking phase where the player tries to reach the "Hall of The Mountain King" while being encouraged to visit three green fireflies that provide hints which will be relevant for a later part of the game. As the player walks around in this phase, distractor enemies will appear once they hit the maximum safe distance away from the centre. These enemies need to be defeated to progress and award experience points that the player can use to either upgrade the size of their contrabass shield or the damage that their conducting baton does. It should be noted that the fireflies fade away during battles to prevent their salience from potentially affecting the attention of the player since this could result in lower concentration [33].

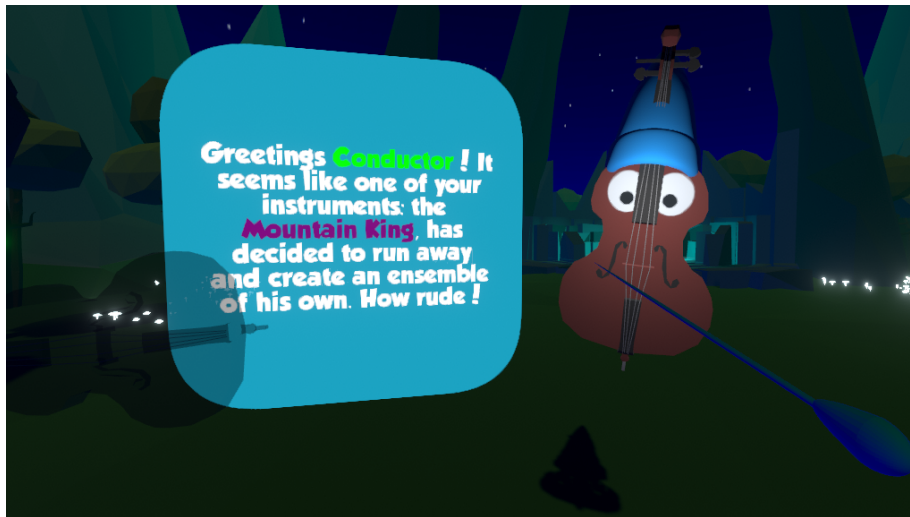


Figure 12: This screenshot from Ensemble Retriever illustrates the tutorial that one of the in-game characters gives to the player.

The walking phase of Ensemble Retriever is finished once the player enters the portal to the "Hall of The Mountain King".

Once the player enters the "Hall of The Mountain King", any redirection gains are disabled. The reasoning for disabling gains is to allow the player to spend the last few minutes of the game getting used to normal head rotations before taking off the HMD. Inside the hall, the player has to answer a multiple choice quiz which asks questions based on the previous hints that could have been acquired from fireflies in the walking phase. The quiz itself cannot be failed, but the final score that the player is given ties with how many correct answers they give. The quiz itself has three questions in total.

Once the quiz is finished, the player will have to fight against the "Mountain King". This is a longer battle where the "Mountain King" will use combinations of all previous attacks that distractor enemies have been using to challenge the player. Once the "Mountain King" has been defeated, the game is finished and the player will be shown their scores which consists of four components:

1. Time Score (How long it took to finish the game. Shorter times means higher scores).
2. Damage Score (How much damage the player has taken in total. Less damage means higher scores).
3. Quiz Score (How many correct quiz answers the player has gotten).
4. Total Score.

A full playthrough of Ensemble Retriever by the author can be seen on YouTube [52] for further illustration.

4.10 Fully Integrating Distractors with Game Mechanics



Figure 13: These are the five distractor enemies that are employed in the Ensemble Retriever game.

The development resource budget of Ensemble Retriever was reasonably small (~1.25 months). As such, in order to have a reasonable scope and allowing for as much code/asset reuse as possible, the main mechanic that was decided to fully integrate with distractors was the enemy encounters in the game. Since the AC2F algorithm relies on rotation gains, the goal of using distractors in this case was to make the player move their head around as much as possible. Ensemble Retriever currently consists of five enemy distractors which can be seen in Figure 13. If we look back to the taxonomy of distractors which was presented in Section 2.3.1, these can be considered as explicit, concrete distractors that are fully integrated into the experience. In this case, the distractors are considered explicit as the player is told they have to fight them.

From a game design point of view, their use can be seen as "random" enemy encounters which is a common mechanic in role-playing games. Instead of being random though, these enemy encounters trigger once the player has reached the maximum safe distance away from the centre of their physical tracking space.

Once an enemy encounter starts, one of the five possible distractors is randomly chosen from a list. The chosen distractor is then removed from this list so it cannot be chosen again for some time. Once all distractor enemies have been chosen from the list, it is then repopulated with all five again. In the worst case scenario, this means that the same distractor might show up as the next enemy right after the list has been repopulated. The rationale for this approach is to avoid situations where

the same distractor can show up too many times in a short space of time, potentially annoying the player due to limited variety as mentioned by Sra et al. [19].

Each of the five distractor enemies rely on projectile attacks that have three potential movement speeds as well as a unique property per type of distractor. Before attacking, the distractor will make use of two telegraphs: one for the type of projectile it will attack with and one for how fast it will be. These telegraphs make use of both animation and audio cues, allowing the player to learn and identify the different types of attacks that are used so they can prepare accordingly.

A standard enemy encounter consists of two phases: a tutorial phase and a proper battle phase. During the tutorial phase, the distractor will use its different attacks in order to teach the player its capabilities. In this phase, the distractor will not move away from its initial position. Once the player manages to counterattack the distractor, the real battle begins. During the battle phase, the distractor will randomly choose an angle between -360 and 360 degrees from the player after each attack, and rotate around them. The attack order will at this point also either be random or predetermined depending on the type of distractor. These approaches force a fair amount of head rotation from the player as they need to keep track of where the distractor moves and where their projectile attacks are. Furthermore, these approaches also makes good use of the full 360-degree space that virtual reality allows while providing many opportunities for applying rotation gains.

The following sections will describe and detail each distractor that exists in Ensemble Retriever. It should be noted that the screenshots of these were taken from the debug mode of the game, hence why the conductor baton and shield are in fixed screen positions.

4.10.1 The Contrabass



Figure 14: This screenshot shows off the contrabass distractor and its projectile attacks which travel directly towards the player.

The first among the five distractor enemies is also the simplest one: "The Contrabass", as seen in Figure 14. It attacks the player with projectiles that move towards them in a straight line. As a means

to make the player move their head around more, the contrabass will mix slow and fast projectiles. A fast projectile will in this case move fast enough that it hits the player before a previously thrown slow projectile. This way, the player has to keep track of where the contrabass itself is when it uses fast projectiles while trying to remember where the slower moving ones are so they can absorb these as they come close.

4.10.2 The Oboe

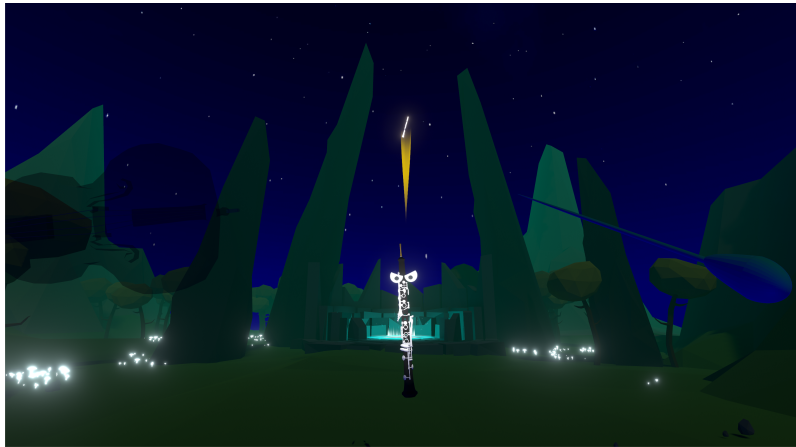


Figure 15: This screenshot shows the oboe distractor and its projectile attack which will travel on a vertical arc above the head of the player.

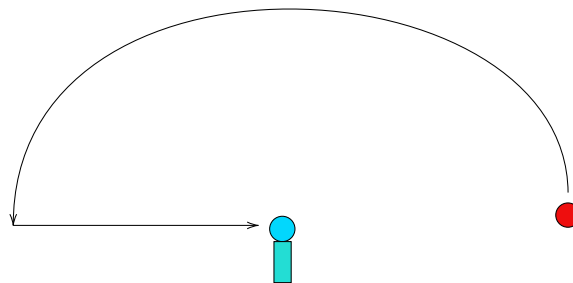


Figure 16: An illustrated, sideways 2D example of how the oboe projectile travels. The blue circle and quad represents the player while the red circle represents the distractor that is about to attack.

The next distractor is "The Oboe", which can be seen in Figure 15. It attacks the player with projectiles that move in a vertical, 180-degree arc above the player's head before travelling straight towards their head. A rough two-dimensional figure illustrating this can be found in Figure 16. The Oboe attempts to facilitate head rotation by trying to hit the player from behind with projectiles. A clever player might end up holding the shield behind their head to absorb these, lessening the

impact of the approach though. Despite this, it is still necessary to keep track of the projectiles as the Oboe moves and rotates around the player.

4.10.3 The Harpsichord

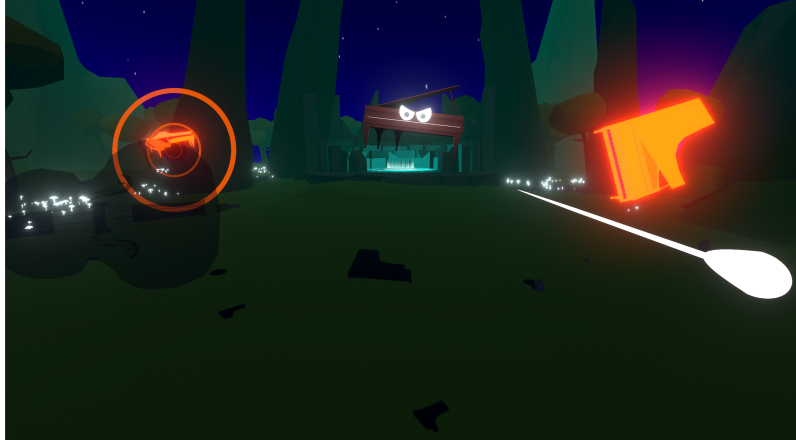


Figure 17: This screenshot shows the harpsichord distractor. It attacks with multiple rapid projectiles from two locations which are further away from its body.

”The Harpsichord”, which can be seen in Figure 17 is the third distractor in Ensemble Retriever. It attacks the player with rapid projectiles that spawn from two wormholes that are further away from its body. This attack method means that the player needs to move their shield from left to right to quickly block incoming projectiles. It allows for some extra head rotation as the player might need to slightly shift their head to clearly see each projectile.

4.10.4 The Violin

The fourth distractor in Ensemble Retriever is ”The Violin”, which can be seen in Figure 18. It is similar to the Harpsichord in the regard that it rapidly fires projectiles. Instead of firing projectiles from two locations though, it instead fires a line of projectiles that the player needs to block with their shield. This approach allows for some additional head rotation as the player needs to trace the line of projectiles that moves towards them. The line itself is long enough that some shifting of head orientation is beneficial to see everything as it gets close.

4.10.5 The Glockenspiel

Finally, we have ”The Glockenspiel”, which is seen in Figure 19. This distractor attacks the player by throwing projectiles that travel on a curve towards the player. The curve itself will intersect with the player at a roughly 90-degree angle to their left from where the projectile was fired. This approach facilitates more head rotation as the player needs to track the path that the projectile flies through before blocking.

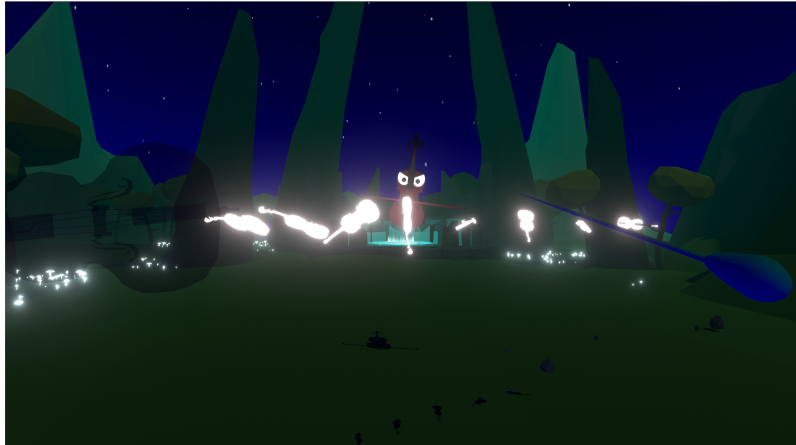


Figure 18: This screenshot shows the violin distractor. It attacks with several projectiles at a time in a line formation.

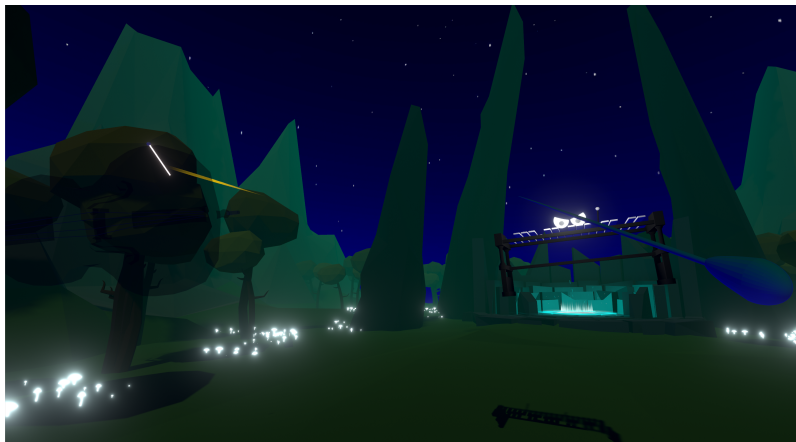


Figure 19: This screenshot shows the glockenspiel distractor and its curved projectile attack. The projectile itself travels on a arc towards the player's left side and will hit them at a ~ 90 degree angle.

4.11 Employing Context Sensitive Reorientation: Teleporters

While distractors are one example of context-sensitive reorientation, they are not necessarily the only ones that exist. There are many other potential options that can reorient the user while still being context sensitive. This thesis in particular makes use of one such method: reorientation when using teleporters. In games, when using a teleporter it is common to be reoriented in a manner so that they will face towards the future direction they are expected to walk. If we slightly modify this concept, we can make use of it for reorientation. As the player enters a portal or teleporter, their facing direction after being teleported is changed in a manner that results in their future expected path to be towards the room centre. As an example: in the Hall of The Mountain King, the player will be teleported to the edge of the room, meaning that there is only one direction they can walk in. By instantly reorienting the player during this teleportation it is possible to guarantee that this future direction will be through the centre of the physical space.

In a way, this could be seen as similar to Suma et al.'s change blindness redirection [26], although it does not require the virtual world to dynamically change. The only thing that changes before and after teleportation in this case is the orientation of the user. Of course, this approach is not very useful if the teleporter sends the user to an open space as the future walking direction could vary. This reorientation method would most likely be at its most useful for more narrow spaces where potential walking directions are limited. In any case, it is a useful tool to consider as an alternative to only using distractors or other resetters. In the future, we may have many different types of context-sensitive reorientation methods outside of just distractors which all could exist together to create more varied experiences. For example, a cylindrical elevator which limits visibility could be used as a context-sensitive reorientation. In this case, it could reorient the user by having a door appear which forces the user to travel through the room centre when leaving. As Sra et al. already have discussed [19], anything that obscures or limits visibility in one way or another can be used to reorient or redirect a user.

4.12 Disabling Redirected Walking Towards the End of an Experience

One particular detail that is further worth clarifying with Ensemble Retriever is that the redirection gains are disabled once the Hall of the Mountain King is entered. The reasoning behind this is so that participants can get used to normal head rotations before taking off their HMD and finishing the experience. Given that adaptation towards redirection is possible [21, 22], it might be ideal to allow users to adapt back towards normal head rotation before they take off their HMD to limit any disorientation symptoms. How effective this approach is on the other hand, has not been measured and would need to be further tested in future work. Regardless, it is something that likely should be considered when developing redirected walking experience to ease the user back into how real head rotation functions. This approach might not necessarily be as easy to integrate with every solution, but it works well in the case of Ensemble Retriever. Since the final battle with the Mountain King occurs in a space where little movement is needed, there is also little need for redirection.

4.13 Providing a Distance Magnitude Cooldown on Distractor Triggers

One small detail that is worth to mention in Ensemble Retriever is the cooldown for triggering a new distractor after one has been defeated. A common way to handle this would be to have a timed cooldown. This would avoid immediately triggering a new distractor if the player is standing roughly on the bounds of the distractor trigger before starting to move again. If the player is standing still on this bounds for a longer amount of time though, this solution will not be effective. Instead, the player is required to walk a set amount of distance before a new distractor can trigger. This is not necessarily an ideal solution either as the maximum safe distance needed to travel before triggering a new distractor would vary depending on room size and require individual calibration. Furthermore, there are some additional issues which are mentioned in Section 7.3. The current implementation with a distance cooldown is the best solution which could be thought of for this thesis, but there are likely better alternatives which could be considered.

Now that the more technical details of Ensemble Retriever have been disclosed, it is time to move on to the first experiment which focuses on the noticeability of distractors.

5 Experiment 1: Noticeability of Distractors

This chapter consists of the Method, Results and Discussion components for Experiment 1, which focuses on measuring detection thresholds for two different distracted states: walking and fighting.

5.1 Method

Before delving into the deeper details of the methods that have been used for Experiment 1, it might be best to give a brief overview of the structure of it first. Experiment 1 consists of having participants play through the implemented Ensemble Retriever VR game, which is detailed in Chapter 4. As they play, the strength of rotation and curvature gains is gradually increased. Once the gains have reached a value that is high enough for the participant to notice, they have to press a button on their controller. This button press records a detection event for the relevant gain that has been applied. The strength of the detected gain is then decreased so it can rise again until the next time it is detected. After finishing, participants answer a short demographical post-test questionnaire with some optional qualitative feedback.

The following sections detail the methods that have been employed to make this possible and what the focus of the measurements is.

5.1.1 Null Hypothesis

When it comes to estimating detection thresholds, one interesting comparison would have been to see how different detection thresholds are between an undistracted and distracted state. While this would have been an interesting comparison, it does not necessarily fit all too well with the concept of fully integrating distractors into the experience. Since the distractors are such an important aspect of a game when integrated, it is not possible to remove them and retain the same experience. This tight coupling makes it challenging to compare distracted and undistracted conditions from a validity point of view as the experience would be significantly different without distractors. Instead, in order to keep the same experience for all participants, this experiment focuses on comparing two different distracted states. These two states consist of the walking state and the battle state in the game.

While walking around in Ensemble Retriever, the player is distracted to some degree as the act of simply playing the game itself would be considered as an abstract distractor. There are some elements like the fireflies in the environment that could be considered as concrete distractors as well. Furthermore, there are a few other elements that contribute to an increased cognitive load and additional distraction in this state. These consist of the working memory task in the game (remembering the clues that fireflies provide), some basic path planning and being prepared to press the detection button as soon as the player notices any redirection. The battle state on the other hand, primarily distracts the player through the use of concrete distractors which consist of

the distractor that the player has to fight as well as the projectiles that it fires. This state is also where most of the challenge in Ensemble Retriever lies as players need to figure out how to deal with the various projectile attacks and counterattack properly.

By having these two interlinked states in the game that each participant takes part in, the experiment itself can be considered as using a within-subjects style of design. Given that the players move very little during battles, it is primarily rotation gain thresholds that are the most relevant to compare between these two states (curvature gains are only applied when walking). Since distraction/engagement is identified as a detection threshold variable in Chapter 2, it would thus be interesting to see what the differences in rotation gain thresholds are between the two states. As such, the following null hypothesis has been established:

E_1H_0 : The mean detection thresholds for the battle state in Ensemble Retriever are not significantly wider than the walking state.

The expectation for the measurements is that that the larger amount of distraction that results from battles should in turn result in it becoming harder to notice redirection.

5.1.2 Estimating Detection Thresholds

In order to test the previously mentioned null hypothesis, it is first necessary to understand how the thresholds are estimated. As already mentioned in Section 2.7, the standard procedure for estimating detection thresholds has a high risk of cybersickness, which makes it unfeasible to use from an ethical point of view. The procedure can also be seen as incompatible with the concept of fully integrated distractors, meaning that a different approach is needed for this experiment.

Instead, this experiment makes use of an alternative approach which is inspired by one of the experiments that Fuglestad performed in his redirected walking research [6]: Incrementally increasing gains until they are noticed. This approach should minimise the risk of potential cybersickness as the possibility of reaching beyond the comfort/cybersickness threshold for individuals is lower. It should be noted that since the estimation method is different, it may of course not be as accurate as the standard procedure from Steinicke et al.'s research [2]. The use of estimation method should in this case be considered as a variable that may affect the estimated thresholds in addition to the previously mentioned variables in Table 1.

Incremental Gain Increases

The incremental increase in gains is handled in two steps:

1. Before each increment, a time step within the range of 2.5-7.5 seconds is randomly chosen. Once the duration of this chosen time step has expired, a random gain will be chosen for incrementing.
 - It does not make sense to accumulate curvature gains when the user cannot experience them. As such, the potential options to increment vary somewhat depending on the state of the game:
 - In the walking state: curvature, negative rotation and positive rotation gains are possible

options.

- In the battle state: negative rotation and positive rotation gains are possible options.
2. Whenever a gain is incremented, it is done at a random amount within a given range:
- Rotation gains are incremented by $\pm 2.5\text{-}7.5\%$ each increment. Only one type of rotation gain is incremented each time.
 - Curvature radius is decreased by $2.5\text{-}7.5\text{m}$ each increment.

All of these random choices are handled through the use of a Mersenne Twister library [43]. In this case, an SFMT implementation [53] has been used to generate random integers corresponding to each choice.

Once the experiment starts, all gains start at a default value. This default value is 0 for rotation gains and a 23m for the curvature gain radius. The reasoning for 23m as the default value for curvature radius is since Steinicke et al. initially found 22m as the threshold for curvature gains in their study [2]. Other studies have generally found lower values as the threshold, but 23m is set as the starting point as it is right above the highest estimated threshold by any studies seen in the literature sample for this thesis.

One potential issue that can arise with this type of method is that the participant does not move their head or body for some time. This results in an unnoticed accumulation of gains which might be uncomfortable and in the worst case result in cybersickness. To decrease the risk of large gain accumulation, a variety of caps have been introduced for each type of gain which cannot be exceeded:

- Negative rotation gains cannot go lower than a value of -0.75. In this case, a gain of -1 would counteract any rotation on the y-axis and give the impression of this axis being frozen. If we use Steinicke et al.'s rotation gain semantics [2], this would be the equivalent of a 0.25 multiplier.
- Positive rotation gains cannot exceed a value of 1. This is the same as the strength that is applied when using the 2:1 Turn reset method. If we use Steinicke et al.'s rotation gain semantics [2], this would be the equivalent of a 2.0 multiplier.
- Curvature radius cannot go lower than a value of 2.5m. A radius of 1m is the lowest that the Redirected Walking Toolkit can use in this scenario. From the sample of literature which was used in this thesis, no curvature thresholds below 3m were found. The closest would be Steinicke et al. mentioning that a curvature radius of $\sim 3.3\text{m}$ was slightly noticeable while still useable [2]. As such, a cap of 2.5m was deemed reasonable for the sake of mitigating cybersickness.

Given the usage of caps in the experiment, there may be several detections which reach these limits during certain situations. In these cases, capped detection values point towards the possibility of being able to further increase redirection, but that the gain has been capped to avoid possible increases in cybersickness.

Detection Events

Once the participant presses a button on their controller to signify that they notice they are being redirected, the following procedure takes place in the background:

- The last 0.5 seconds of applied gains are always stored by the software.
- Once the detection button has been pressed, the gain that has had the largest prevalence in this buffer is chosen as the detected gain.
- The detected gain is then halved for rotation gains or increased by 25% for curvature radius as a means to "reset" while also allowing gains to quickly rise back towards the next detection event.

Already established methods like Steinicke et al.'s [2] make use of multiple buttons where the users for example can indicate whether they noticed that their head rotation was larger or lower than in reality. The reasoning for instead opting for a one button procedure is that it should be as simple as possible for the user to notify that they have detected something. In the case of standard estimation procedures, the entire focus of the participant is on the experiment itself, making multiple buttons feasible. In the case of this experiment though, the user is playing through a game and it would not be ideal to require additional mental processing when they already are engaged and focused on something else. Doing otherwise would detract from the point of using distractors as the focus of the user primarily should be on those.

5.1.3 Performance Data Collection

As the participants play through Ensemble Retriever, the software itself collects a large variety of analytical data per frame. The general thought process behind recording data every frame is to provide insight into the whole runtime of the experiment and allow for deeper analysis if necessary around detection events. This data collection consists of data like the strength of gains at any given frame, whether a distractor was active, which distractor was active, whether a gain was detected and so on. The full list and a description for each piece of data that is collected can be found in Section B.2.2 as it is too long to reasonably mention in this section.

5.1.4 Participant Sample

This section consists of information on how the acquired participants were sampled, an overview of the demographic in the sample, what information participants were given and miscellaneous information.

Sampling Procedure

The sampling procedure for this experiment consisted of the following methods:

1. The experiment was advertised by supervisors during lectures.
2. Posters advertising for the experiment were posted around campus.
3. Advertisement on the local NTNU Discord¹ server.
4. Convenience Sampling.

¹<https://discordapp.com/>

Participants were given access to a doodle poll² where they could choose between a variety of time slots for when they wanted to partake in the experiment. Participants made use of aliases for this sign-up process. As compensation for participating, all participants received some chocolate. In total, 22 participants took part in Experiment 1. One of these includes the author of the thesis.

Sample Demographic

In order to provide some demographic insights, participants were asked to answer a post-test questionnaire. The demographic questionnaire that participants answered can be found in Appendix C.

The sample itself primarily consisted of students within the age range of 18-24 years. 22 of the participants identified as male while 2 identified as female. Among the participants, only one needed to remove any optical corrections to use the HMD. In this participants' case, it should be noted that this was their preferred way to enjoy VR and did not negatively impact their vision.

7/22 participants have had prior experience with projects using redirected walking. Finally, Figure 20 shows a histogram over the Likert scale answers for the prior VR experience of participants.

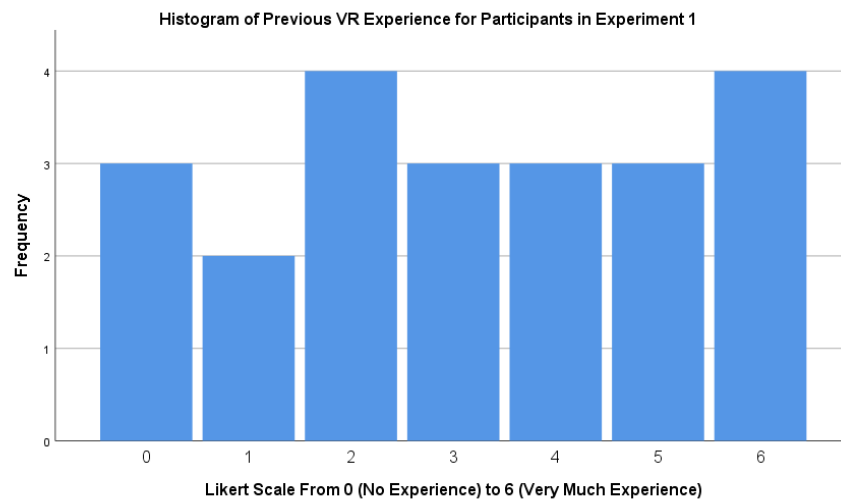


Figure 20: This histogram shows the frequencies of what values participants provided in the demographic questionnaire of Experiment 1 in terms of prior VR experience.

Information and Consent

In terms of information and consent, participants were given an information and consent sheet where written consent was necessary. This information sheet was also available to read during the sign-up process. The information/consent sheets can be found in Appendix D for additional details. Furthermore, participants were also given some oral information before playing through Ensemble Retriever. This information can be summarised as follows:

- A very brief summary of what redirected walking is was given if necessary.

²<https://doodle.com/>

- If the participant experienced any form of cybersickness at any point they were instructed to mention this so that the experiment could be stopped.
- Participants were encouraged to not walk too quickly for their own safety's sake.
- Participants were informed that the room tracking might fail for the controllers at times. If so, they need to wiggle it in the air for the problem to resolve.
- Participants were told they should be mindful of the tethered cable.
- Participants were told that they should not move too much around during battles to avoid potentially moving into walls.
- Finally, participants were recommended to always keep a finger on the detection button so they could press it as soon as they noticed any redirection.

The specific details and wording that was used for this information can be found in Section [B.2.1](#).

Cancellation, Interruptions and Miscellaneous Information

One of the participants had to cancel the experiment due to cybersickness. It should be noted that this participant mentioned they could easily get motion sickness which cybersickness can be considered a form of [23]. As such, it likely had some effect on why this was the only participant needing to cancel the experiment. This participant also mentioned that they mistakenly pressed the detection button several times. Due to this, they have been excluded from the sample, resulting in a sample of 21 participants.

Two participants in the experiment did not manage to fully finish Ensemble Retriever due to the following reasons:

1. One participant got interrupted by a fire drill towards the end of the game.
2. One participant managed to disconnect the cables between the HTC Vive Linkbox and HMD towards the end of the game.

The data for these two participants is still included in the sample as the interrupting moments happened late into the experiment.

Finally, four participants did not provide any detection data at all. Two of these did not provide any detections as they did not notice being redirected. These were participants with no prior VR experience and as such, they mentioned that they had no prior reference points to use for detecting any unusual manipulations. The other two appear to have misunderstood which button to press on their controllers for detection despite the tutorial clearly labelling and showing which one to use.

5.1.5 Experiment Environment/Physical Space

The experiment itself took place in a room with a physical tracking space of approximately 5m x 5.75m and can be seen in Figure 21, Figure 22 and Figure 23. The lights in the room were turned off during the experiment to prevent any potential reference points to the floor which might have been gleaned from under the HMD. Doing so is a relatively standard procedure as Steinicke et al. made use of this in their initial detection threshold experiment [2].



Figure 21: These two images show a rough outline of the physical tracking space that was used with green lines. Red circles are used to outline the location of the Vive Lighthouses.



Figure 22: This supplementary image shows more of the physical tracking space.

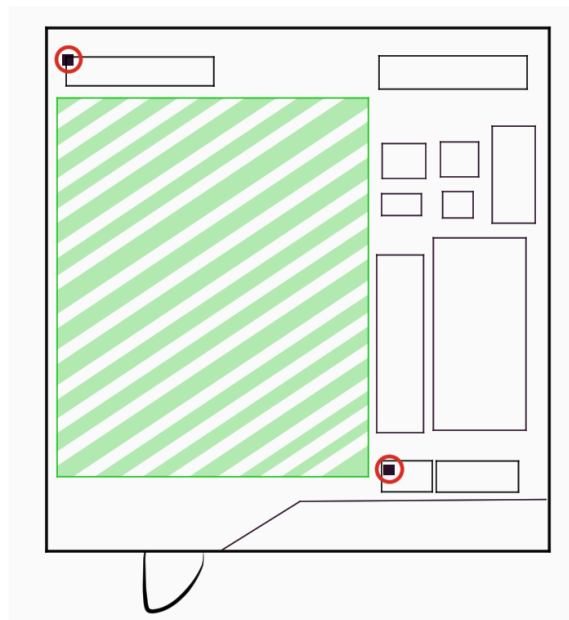


Figure 23: This sketch represents roughly how the experiment room was used from a top down view. The green shaded area is the physical tracking space while the location of the Vive Lighthouses is signified with red circles.

5.1.6 Data Post Processing

In terms of rotation gain detections, it is not uncommon to find detections in the recorded data that have been set to a value of 0. There are four potential scenarios where this can happen:

1. The player noticed a gain, but by the time they pressed the button they had already been aligned and gains were disabled.
2. The player noticed that rotation gains were disabled after they had been aligned.
3. The player pressed the button thinking they had noticed a gain while they in reality did not experience any.
4. The player pressed the button a bit late, during a time where their head did not move. This can create the assumption that not applying gains was the most prevalent in the 0.5-second buffer.

Among these four, it is primarily cases 1 and 4 that would be the most important to deal with as some additional processing around these could yield additional detection data. As such, a post-processing step was conducted to attempt retrieving the correct values for these cases. The details on how the post-processing was conducted can be found in Section [B.2.3](#).

It should be noted that the post-processing step for the first 13 participants could only be partially conducted. The reasoning for this was due to a bug in the data recording that affected the validity of certain variables which were aimed to be used for post-processing. In particular, there was an issue where a set of ratio variables representing the ratios of gains applied in a specified time buffer before each detection was not recorded properly. As a result, the only post-processing that was possible for these participants was alternative 1 for case 1 which is described in Section [B.2.3](#). This bug was fixed for all subsequent participants and as such, the full post-processing was used for these.

5.2 Results

This part of the chapter provides the results of the experiment related to the established H_0 hypothesis as well as insights from demographical data. Data visualisation for these results is provided by SPSS due to the large data file sizes. A significance level of 0.05 was used for all statistical tests. For the sake of disclosure for any graphs that directly mention participant ID's, the author's ID is 16. It should also be noted that the participant with ID = 10 is excluded from the sample due to prior mentioned reasons. In terms of presentation, the majority of the graphs will be comparing the S2C and AC2F algorithms as they respectively represent the walking and battle states in Ensemble Retriever.

It should be noted that this part of the chapter only consists of the experiment results. Discussion around these results can be found in Section [5.3](#).

5.2.1 Rotation Detections

As the main focus of Experiment 1 was on detection thresholds, these first few sections are dedicated to presenting the graphs over detection events for participants as they played Ensemble Retriever.

Detected rotation gains are initially presented with curvature radius detections following this.

Raw Rotation Detections Sample

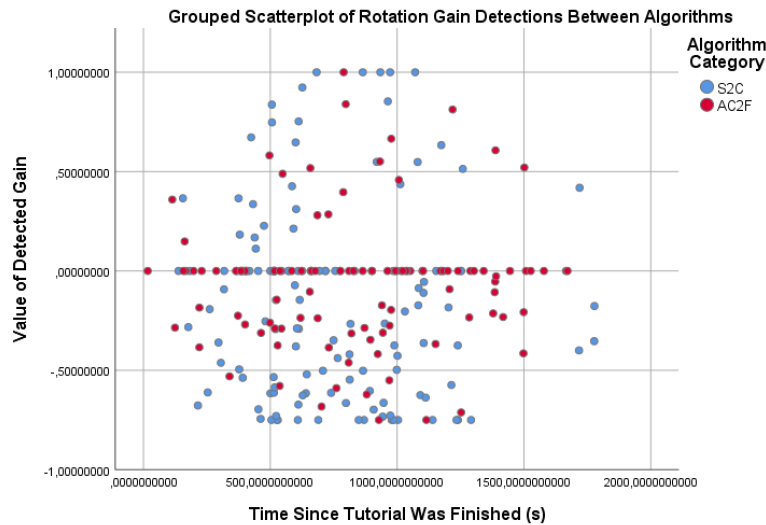


Figure 24: This scatterplot shows the raw data, including outliers for rotation detections between the two employed algorithms. S2C is used for the walking state while AC2F is used for the fighting state.

Before seeing the post-processed sample of rotation gain detections, it might be interesting to first see how the raw data looks like. This raw data can be seen in Figure 24. The various cases where detected gains are at a value of 0 is what the post-processing step aimed to minimise.

Finalised Rotation Detections Sample

The post processing step which was mentioned in Section 5.1.6 resulted in 52 additional rotation gain detection events. This addition results in a total of 213 rotation gain detections throughout the entire sample. The processed data can be seen in Figure 25, where it is grouped by algorithm and Figure 26, where it is grouped by participant ID. To provide some additional visualisation, a line chart showing the progression of detections between participants can be seen in Figure 27 and Figure 28.

A figure illustrating the incremental progression of rotation gains over time for participant 16 (the author) can be seen in Figure 31. Finally, individual scatterplots for negative/positive rotation gain detections with an included regression line can be found in Figure 29 and Figure 30. It should be noted that some participants are not shown in these graphs as they did not provide any detection events.

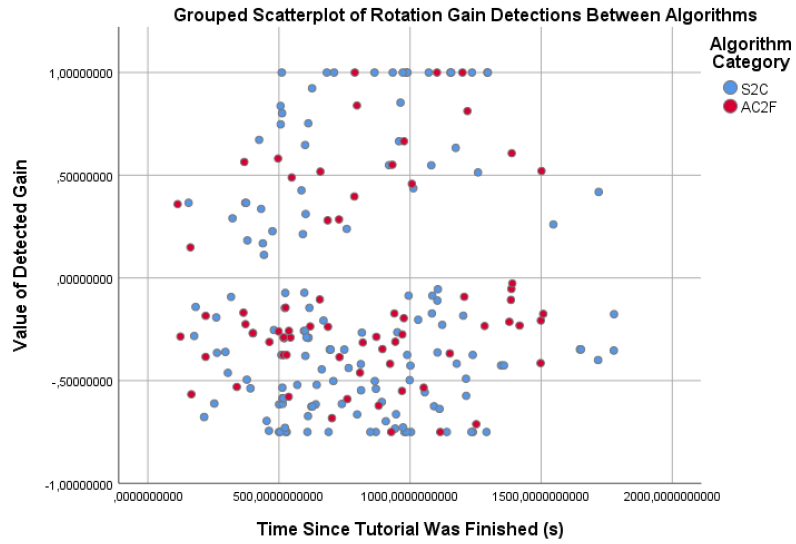


Figure 25: This scatterplot shows the finalised data of rotation gain detections and is grouped by algorithm.

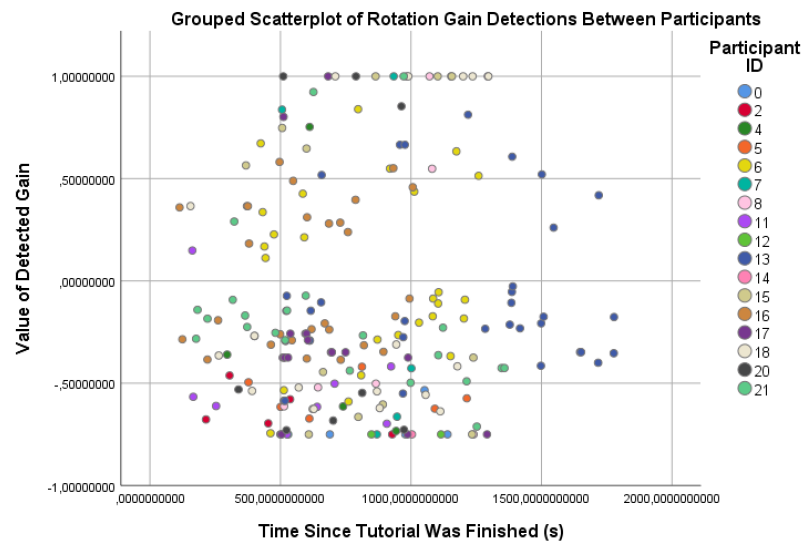


Figure 26: This scatterplot shows the finalised data of rotation gain detections and is grouped by participant ID. Some ID's are not present as these participants did not notice any redirection or misunderstood the task they were given.

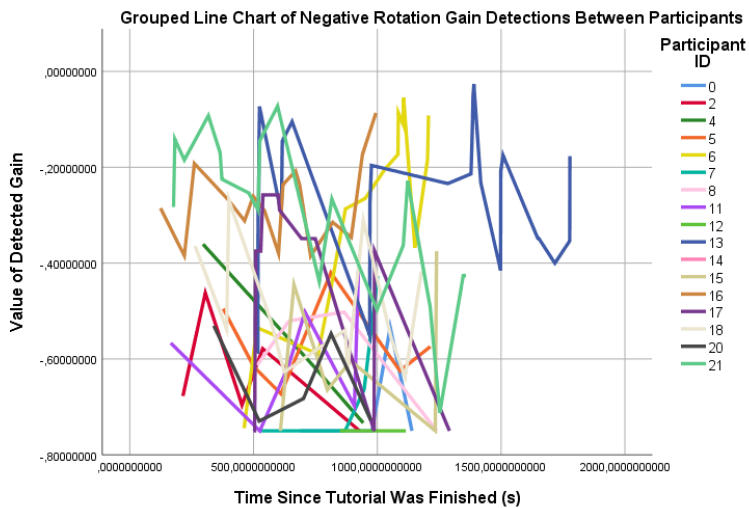


Figure 27: This line chart shows the progression in negative rotation gain detections between participants.

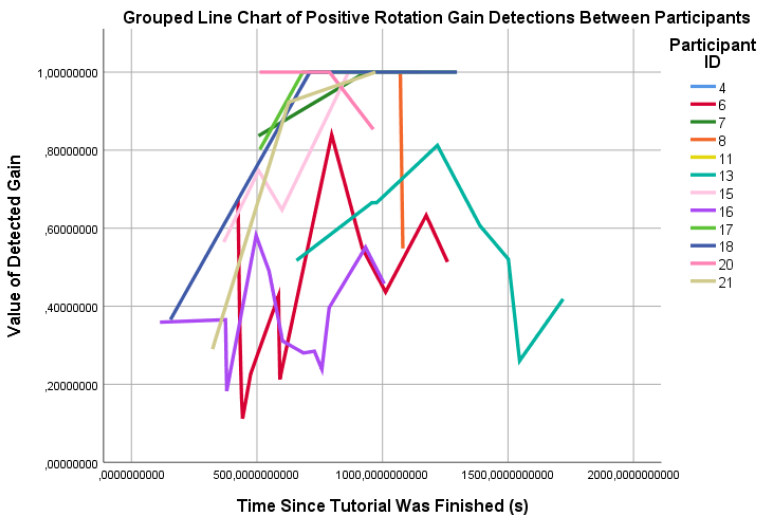


Figure 28: This line chart shows the progression in positive rotation gain detections between participants.

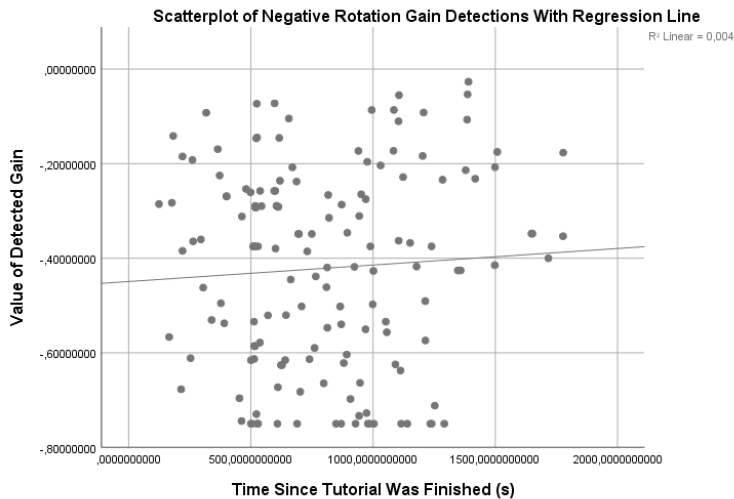


Figure 29: This scatterplot shows the spread of negative rotation gain detections with an included regression line.

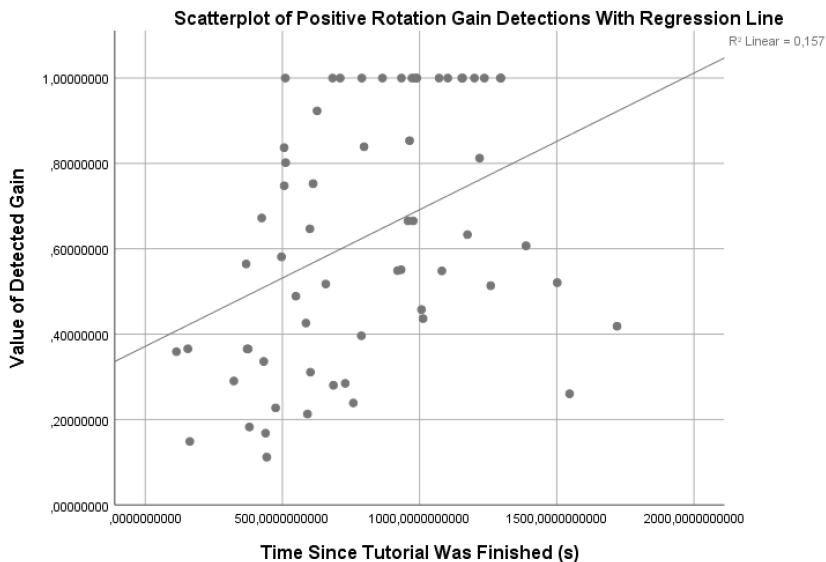


Figure 30: This scatterplot shows the spread of positive rotation gain detections with an included regression line.

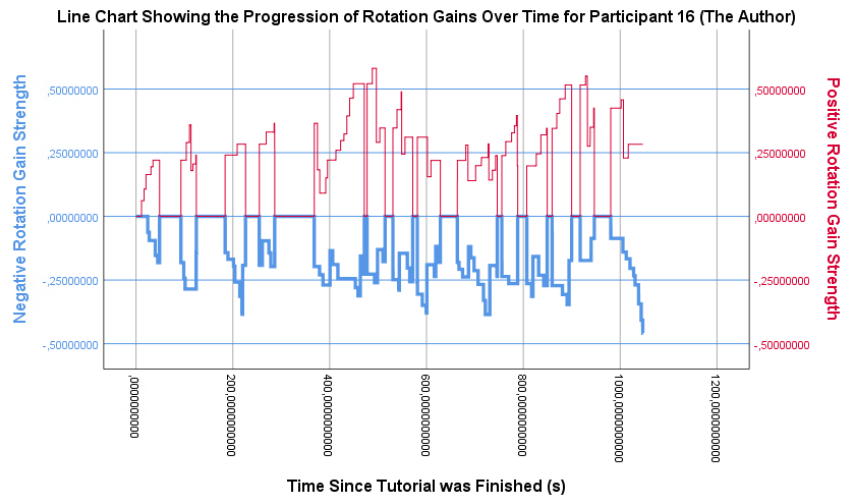


Figure 31: This line chart shows the progression of rotation gains over time for participant 16. Rotation gains gradually increase over time until they are detected, after which they are dropped by 50%. Sections of time where gains are 0 are during distractor battles where the future virtual path has been aligned with the physical room centre.

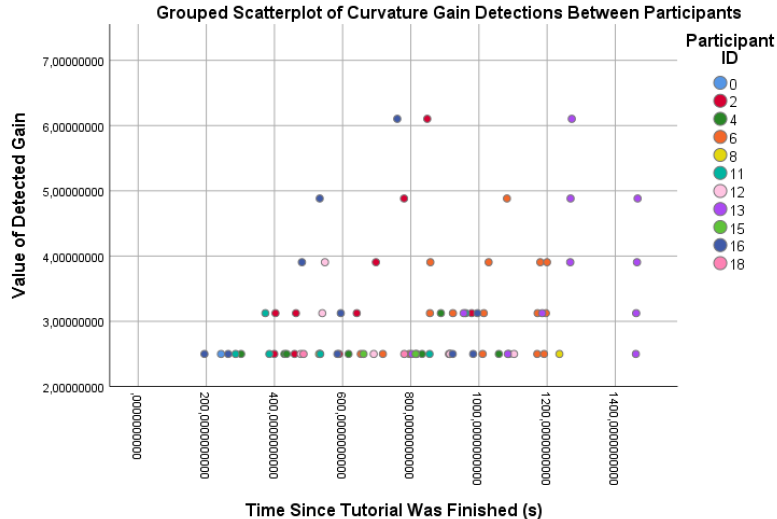


Figure 32: This scatterplot shows the finalised data of curvature gain detections and is grouped by participant ID. Curvature values in this case are defined as the radius of the curvature arc.

5.2.2 Curvature Detections

The various curvature gain detection events that happened during the walking phase of Ensemble Retriever can be seen in Figure 32.

5.2.3 Mean Detection Thresholds

Type of Gain	Mean Detection Threshold	N (Total Number of Detections)
S2C: Positive Rotation Gain	0.6479	N = 43
S2C: Negative Rotation Gain	-0.4631	N = 101
AC2F: Positive Rotation Gain	0.5828	N = 19
AC2F: Negative Rotation Gain	-0.3365	N = 50
S2C+AC2F: Positive Rotation Gain	0.6279	N = 43+19
S2C+AC2F: Negative Rotation Gain	-0.4212	N = 101+50
S2C: Curvature Radius	3.0978m	N = 78

Table 4: This table shows the various detection thresholds that were calculated as the mean of all detection events in each respective category.

The aggregated mean detection thresholds which have been generated out of the previously presented data can be seen in Table 4.

5.2.4 Test for Normality and Choice of Significance Test

Before performing any statistical significance tests on the data, it is first necessary to test the normality of it. For this, the Shapiro-Wilk [54] test was used on positive and negative rotation gain detections. There is no possibility to compare different conditions for curvature gains, and as such, no normality test was performed on this data.

Positive Rotation Gain Detection Normality

The Shapiro-Wilk test resulted in $p < 0.001$ for the S2C category and $p = 0.295$ for the AC2F counterpart. The S2C category is thus not normally distributed while AC2F is.

Negative Rotation Gain Detection Normality

The Shapiro-Wilk test resulted in $p < 0.001$ for the S2C category and $p = 0.015$ for AC2F. Both of these are thus not normally distributed.

Choice of Significance Test

Since the data is not normally distributed, it is not possible to use standard tests like the independent samples t-test. Instead, the Mann-Whitney U non-parametric test [55] is used as it does not assume normally distributed data.

5.2.5 Hypothesis Testing

Before performing the Mann-Whitney U test, it is first necessary to determine whether the shapes of the data between conditions is similar or not.

Mann-Whitney U Shape Test

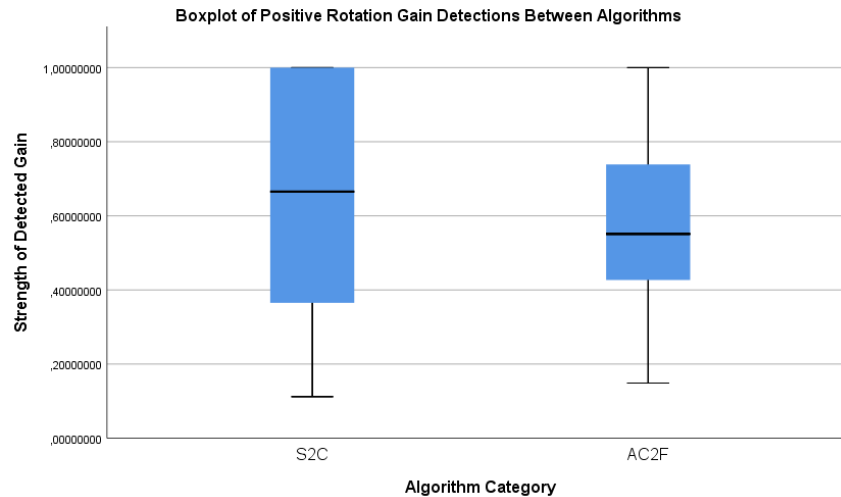


Figure 33: This boxplot shows the spread of detected gains between algorithms for positive rotation gains in Experiment 1.

The shapes of the data can be seen in Figure 33 and 34. In general, it does not seem as if the shape between the conditions is similar enough in either of the cases. As such, the Mann-Whitney U comparison will be on mean ranks. It should also be noted that the Mann-Whitney U test's assumption of independent observations cannot be fulfilled due to the within-subjects design of the experiment.

Mann-Whitney U Results

The results of the Mann-Whitney U provided a value of $U = 362.5$, $p = 0.478$ between S2C and AC2F conditions for positive rotation gains. This means that there are no significant differences between the states for these gains. For negative rotation gains, the statistical test resulted in $U = 1639$, $p < 0.001$, meaning that it is significantly harder to notice negative rotation gains in the walking state in Ensemble Retriever.

5.2.6 Demographic Insights

In order to look for demographic insights, a correlation analysis has been conducted. This analysis focuses on rotation gain detections and makes use of the Spearman's Rho correlation test as the detection data is not fully normally distributed. The correlation matrix can be seen in Figure 35. The most interesting among these correlations is the positive correlation between prior VR experience

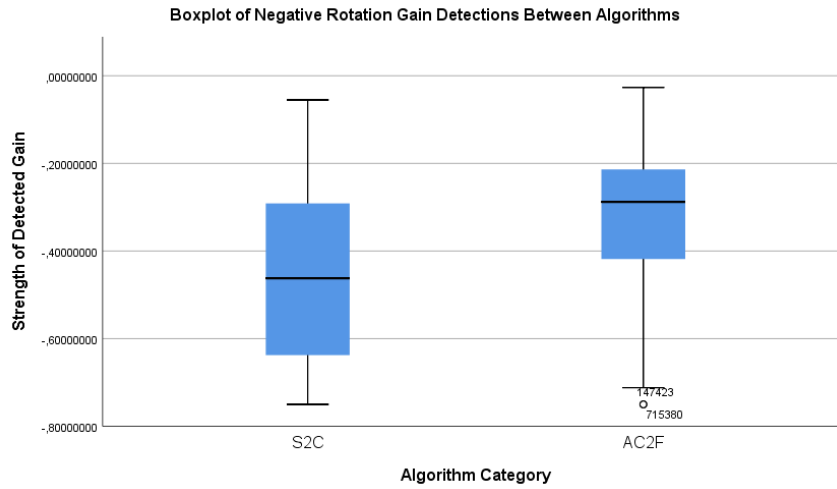


Figure 34: This boxplot shows the spread of detected gains between algorithms for negative rotation gains in Experiment 1.

Correlation Matrix for Demographical Data and Strength of Detected Rotation Gains

		Gender	Age Range	Need to Remove Optical Corrections	Prior RDW Experience	Prior VR Experience	Strength of Detected Rotation Gain
Spearman's rho	Gender	Correlation Coefficient	1,000	,082	-,010	-,129	,204**
		Sig. (2-tailed)	.	,236	,890	,060	,003
		N	213	213	213	213	213
	Age Range	Correlation Coefficient	,082	1,000	,041	,120	-,367**
		Sig. (2-tailed)	,236	.	,556	,081	,000
		N	213	213	213	213	213
	Need to Remove Optical Corrections	Correlation Coefficient	-,010	,041	1,000	-,064	-,007
		Sig. (2-tailed)	,890	,556	.	,353	,921
		N	213	213	213	213	213
	Prior RDW Experience	Correlation Coefficient	-,129	,120	-,064	1,000	-,598**
		Sig. (2-tailed)	,060	,081	,353	.	,000
		N	213	213	213	213	213
	Prior VR Experience	Correlation Coefficient	,204**	-,367**	-,007	-,598**	1,000
		Sig. (2-tailed)	,003	,000	,921	,000	.
		N	213	213	213	213	213
	Strength of Detected Rotation Gain	Correlation Coefficient	,196**	,074	,110	-,015	,151*
		Sig. (2-tailed)	,004	,279	,111	,826	,028
		N	213	213	213	213	213

** . Correlation is significant at the 0.01 level (2-tailed).
 * . Correlation is significant at the 0.05 level (2-tailed).

Figure 35: Correlation matrix for the demographic data in Experiment 1 and the rotation gains that were detected. N values are 213 as there were 213 rotation gain detections.

and the strength of detected gains. As such, a deeper analysis has been conducted in Section 5.3.9 to gain insights as to why this is the case.

Prior VR Experience and Rotation Gain Detections

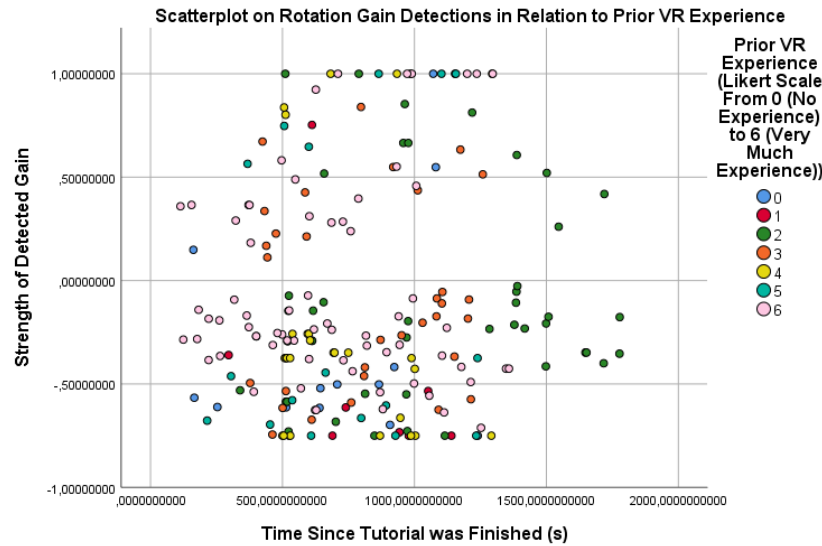


Figure 36: This scatterplot shows the rotation gains that were detected in Experiment 1 and their relation to prior VR experience.

As a basis for discussion in Section 5.3.9, a scatterplot on the strength of rotation gain detections in relation to prior VR experience can be seen in Figure 36. Boxplots for positive and negative gains can be seen in Figure 37 and 38.

Qualitative Feedback

The participants also provided some written qualitative feedback through the post-test questionnaire. This section summarises the feedback that was provided on a question by question basis.

Whether the Participants Found Any Bugs or Glitches

Some participants experienced minor controller tracking problems when standing in certain areas of the physical tracking space. These were quickly resolved by holding their controllers up in the air.

Whether the Participants Found the Experience Enjoyable

In general, participants found the experience to be very positive. Despite this, some mentioned that the experience was rather long, so fighting the same distractors repeatedly got a bit annoying towards the end.

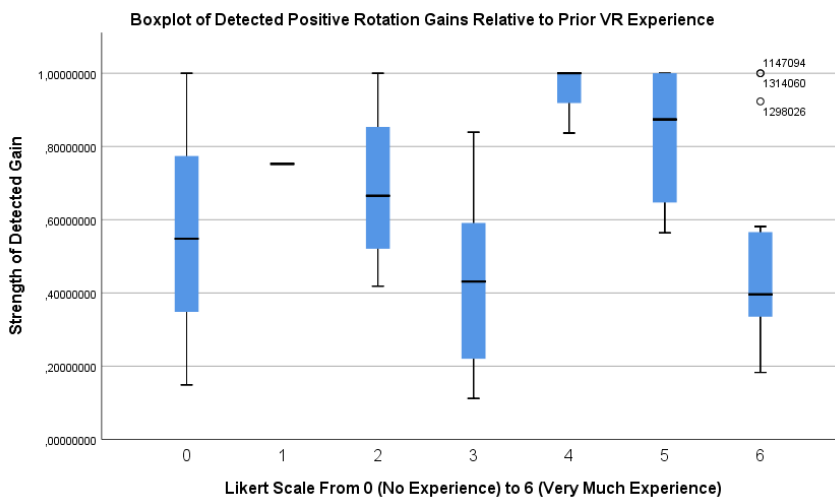


Figure 37: This boxplot shows the spread of detected positive rotation gains in relation to prior VR experience.

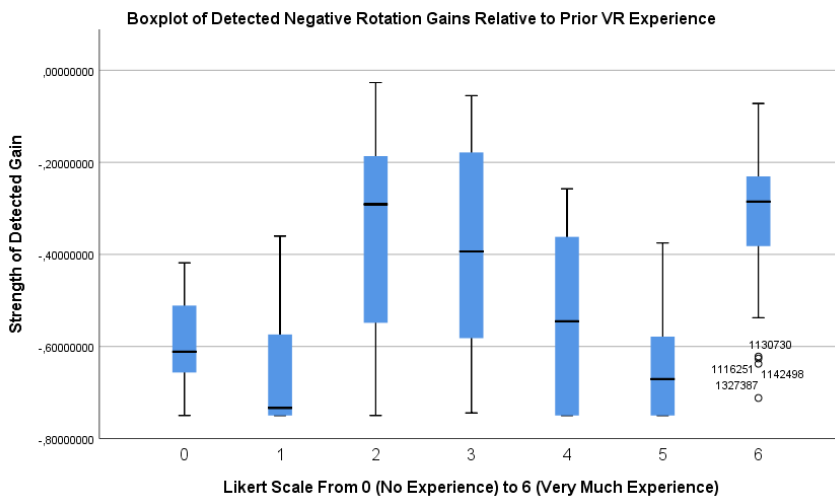


Figure 38: This boxplot shows the spread of detected negative rotation gains in relation to prior VR experience.

How the Participants Felt About the Redirection Techniques

When noticed, participants mentioned that the redirection felt rather odd initially. After a while, some participants got used to the effect and found it to be effective while others mentioned it was a bit awkward and annoying.

Whether the Participants Had Any Problems Throughout Their Experience

In general, participants mentioned that it was rather frustrating to untangle themselves from the HMD cable as they played. Three participants also mentioned that they experienced a mild amount of discomfort due to the redirection, but that they did not feel like it was enough to mention during the experiment.

Additional Comments From Participants

Other than the already mentioned feedback, participants did provide some additional comments on how they felt the experience could be improved. The majority of this feedback has been kept in mind when making changes for Experiment 2. In particular, participants wanted some of the distractors to fire slightly faster projectiles as some of the slower ones became a bit boring over time. The full list of software changes between Experiment 1 and 2 can be seen in Section 6.1.6.

5.2.7 Summary

A contextual summary of how the various variables from Chapter 2 relate to this experiment can be found in Table 5. It also includes some potentially new and relevant variables based on the results that have been shown. In general, the results showed that negative rotation gains are significantly easier to detect during battles (Mann-Whitney U: $U = 1639, p < 0.001$), compared to when walking around. No significant difference could be found for positive rotation gains between the walking and battle states (Mann-Whitney U: $U = 362.5, p = 0.478$).

5.3 Discussion

This section focuses on discussing the presented results in Experiment 1.

5.3.1 Rotation Gain Adaptation Effects

It is interesting to see that there appears to be an adaptation effect for positive rotation gains, but not negative through the results. Research has already shown that adaptation effects can occur for curvature gains [21] and this has provided speculation as to whether this could be the case for other types of gains as well. The primary question lies in why the adaptation effect primarily happens with positive rotation gains and not negative. It may be that disabling redirection gains during alignment towards the centre has some effect on adaptation and as such, might somewhat obfuscate the full adaptation effect. In the case of this experiment, the estimation method might also have contributed to adaptation as it incrementally increases instead of randomly testing a range of gains. In order to more accurately test this adaptation effect, one approach could be to not disable gains during alignment. This approach would result in behaviour where gains dynamically change at a certain point from positive to negative and vice versa. The reasoning for this behaviour

Variable	Context
Size + Shape of Physical Tracking Space	5m x 5.75m rectangle
Optical Flow/Visual Density in VE	Low Poly Artstyle. Limited visual density outside of environmental trees and shining mushrooms. The walkable environment is fully open. The player has to fight enemy distractors while exploring which move around the player and shoot various salient projectiles that might contribute to increased optical flow and visual density. See figures in Chapter 4 for examples.
Hardware: HMD Field of View	HTC Vive. 145 diagonal degrees FoV.
Speed of Walking	~ 0.2576 m/s on average (Total distance travelled / Total time spent walking between participants). Static curvature gains employed.
Engagement/Distraction	The participants played through a VR game which consisted of walking around, collecting clues (working memory task) and fighting enemy distractors while exploring. Overall engagement levels based on feedback and observation of participants could be considered as high.
Awareness of Redirection	Participants were informed of redirection and the basics of redirected walking. The concept of distractors was not directly explained, but mentioned in the information/consent sheet.
Gender	20M, 2F (One among the 22 participants was excluded from the data analysis)
Estimation Method	Incrementally and randomly increasing one gain at random timesteps. Rotation gain is halved when detected, curvature radius is multiplied by 25% when detected.
Adaptation: Curvature Gains	The experiment employed an incremental gains method for estimation. Gains would as such, gradually increase over time. Some adaptation could be possible, but the curvature detection data cannot answer this.
Adaptation: Positive Rotation Gains	The data does appear to point towards a potential adaptation towards positive rotation gains. The variability of negative rotation gain detections makes it hard to say whether this also is true for these.
Prior VR Experience	While relatively hard to directly conclude with the current data sample, participants with the highest amount of VR experience were the most sensitive towards redirection. Those with no prior VR experience were among the least sensitive. It should also be noted that some participants who had no prior VR experience could not detect any redirection as well.

Table 5: Summary of variables from Chapter 2 with some additions and how Experiment 1 relates to these.

is that the algorithms will continuously overshoot and try to correct back towards alignment. As such, there may still be better approaches in terms of how to most accurately facilitate adaptation.

It would be interesting to see future research on how this temporarily disabled gain affects adaptation effects. Another area that would be interesting to look deeper into is whether negative rotation gain adaptation is possible in specific scenarios.

5.3.2 Individuality of Detection Thresholds/Events

When looking at the results of the experiment, one apparent detail that can be seen is how large the detection differences are on an individual level. This detail is in line with results from prior research [12, 20, 18, 6] and should not come as particularly surprising. Despite this, the individuality of detection thresholds should be considered when developing redirected walking experiences. By creating short and effective detection threshold calibrations, it could be possible to provide an optimised experience for individuals. As such, it would be interesting to see further research into calibration approaches that are short enough for users to employ, while effective enough to estimate thresholds with strong accuracy.

5.3.3 Variability and Asymmetry in Detection Data

In Figure 26, we can see that there is a large amount of variability in detection events, even on an individual level. There can be many reasons for this, but there is one in particular that participants mentioned as making it very easy to notice. If participants keep looking left and right repeatedly, it becomes easy to notice that they are being redirected. The reasoning for this is that it becomes simpler to notice that the head does not return to the same orientation between the head turns. This issue is something that the S2C algorithm can mitigate through its dampening component whenever the participant is looking towards the centre of the physical space. AC2F as an algorithm does not necessarily have the same solution, but disabling redirection gains during alignment temporarily stops this scenario from being possible. In any case, this is not a particularly simple problem to solve, but it might be possible to attempt detecting when repeated left/right head turns are used and decrease or disable redirection.

Other than variability, there is also an interesting asymmetry in terms of the number of detections that happen for positive and negative rotation gains. While it is not uncommon for positive rotation gain thresholds to be higher than negative ones in existing research, there is little to no information on the asymmetry of number of detections. This lack of information is primarily due to it not being possible to observe this effect when using the standard estimation method which the vast majority uses. It would be interesting to see further research into why the asymmetry is like this as positive rotation gains are harder to detect than negative ones in this case. This is some idle speculation, but it may be that the human brain is less susceptible to notice a redirection when they overshoot a target rotation, rather than undershooting it. Reaching the target rotation may be vital and as such, result in harder to notice redirection as the additional overshooting could be less important. In any case, further research into this topic would likely require more background within psychology and neuroscience to fully understand the asymmetric perception of gains.

5.3.4 Curvature Gain Detection Patterns

Another interesting part of the results were the patterns of curvature detection events. When looking at Figure 32, we can see a very rigid ladder structure in terms of the detected curvature gains. Upon closer inspection, it is possible to see that this ladder effect comes from having a participant continuously press the detection button until they stop to notice the curvature gains. Each press will increase the curvature radius, and since all of these detections start at the minimum 2.5m radius, this ladder effect happens. The interesting thing is that the curvature gains are only detected when they are at the minimum radius while resulting in multiple detection events afterwards. It may be that simply noticing the curvature gains at one point increases the susceptibility to noticing it for a short time. This susceptibility increase would explain why participants need to go through multiple detection events until it becomes unnoticeable again. The fact that all of these detections start at a curvature radius of 2.5m before the resulting ladder effect is unfortunate as it means the cap for curvature gains was set too high. It might have been more ideal to put it to the same minimum radius as the redirected walking toolkit allows, which would be 1m. Doing so would be a lower risk in terms of potentially resulting in cybersickness than increasing caps for rotation gains. The reasoning for this is that the redirection effect is somewhat less immediate in strength due to slow walking speeds compared to head rotations, which may be rather fast.

5.3.5 Curvature Gain Adaptation

Given that all curvature gain detections start at a curvature radius of 2.5m with some following detections, it is likely that adaptation plays some role in why the detections happen this late. The incremental gains style of estimation method means that curvature gains will gradually increase in strength. This incremental increase has been seen to cause adaptation effects in a study by Grechkin et al. [22] and should be considered when looking at the estimated curvature gain threshold. From a practical point of view when working in redirected walking solutions, it could be useful to maximise the usage of this adaptation. Gradually increasing curvature gains until stopping at a strong, but set threshold could decrease the likelihood of detection. As a result, the effective redirection would be rather strong after some time has passed. The trade-off for this approach is of course that the effectiveness of curvature gains would be decreased until the target gain has been reached. Regardless, this could be a beneficial effect to consider.

5.3.6 Deciding on Which Estimated Thresholds to Use for Experiment 2

Given that multiple thresholds were calculated, it is necessary to decide which of these will be used as the gain strengths in Experiment 2. For positive rotation gains, this is a rather simple choice as there was no significant difference between the walking and battle states. In this case, the mean threshold for all positive rotation gain detections is used (0.6279). For negative rotation gains though, there is a little bit more choice to consider.

While there is a significant difference between the walking and battle states, there is a substantial variability in the negative rotation gain detections. Given this large variability, it is rather hard to conclude how accurate the mean threshold would be. As such, the safest option in this case is to

choose the lowest bounded threshold, which is the threshold for battles (-0.3365). The definition of "safe" in this case is to focus on minimising potential cybersickness symptoms of participants.

Another option would be to dynamically change the strength of gains depending on which state of the game that participants are in. Given the results, this would only make sense for negative rotation gains, but it is something that could be considered. The current redirected walking solution in Ensemble Retriever does not have the functionality for switching between pairs of gains depending on the state of the game. Despite this, it should not be particularly challenging to implement as future work.

5.3.7 Comparing the Results With Fuglestad's Study

Since Fuglestad performed a similar incrementing gains experiment in his research [6], it would be worthwhile to compare his results with the ones from Experiment 1. In his research, Fuglestad employed an abstract distractor where participants had to shoot a variety of moving targets with an ingame gun. Fuglestad's threshold results were provided with Steinicke et al.'s gain semantics [2], so for the sake of comparison, they have been converted to the percentage-wise semantics of the redirected walking toolkit. Fuglestad observed a 0.835 threshold for positive rotation gains and -0.31 threshold for negative rotation gains. In terms of negative rotation gain thresholds, the results are somewhat similar if we consider the -0.3365 threshold, which was seen during battles in Experiment 1. The reasoning for comparing with the threshold for battles is that this game state is the most similar to the abstract distractor that Fuglestad employed. The similarity in this case is that Fuglestad's incremental rotation gain experiment did not make use of any walking. The similarity in thresholds could support the validity of Fuglestad's results, given that the estimation methods in this case also were similar. Furthermore, it could point towards similar effectiveness in terms of holding participant attention through the use of distractors.

The biggest difference is the positive rotation gain threshold as the one that was decided for use with Experiment 2 was a threshold of 0.6279. This value reflects a considerable difference which warrants further discussion. The most apparent difference between Fuglestad's study and this experiment is that the redirection gain caps were set far higher. In Fuglestad's case, these were 4.0 for positive rotation gains and -0.9 for negative. This is likely the largest contributor to this as Fuglestad's results showed many cases where positive rotation gain detections were in the higher end of the scale. This difference in caps would skew the data in a way that is impossible for Experiment 1 as a 1.0 cap was used for positive rotation gains. As a side effect, the participants in Fuglestad's study experienced a fair amount of cybersickness. It should be noted that researching this was intentional though, as Fuglestad searched for differences between detection and discomfort thresholds.

Outside of the differences in redirection gain caps, there are also differences in terms of potential optical flow, tracking space size and potential distraction/engagement, which could have further effects. In any case, it is interesting to see that the results are at least similar in terms of negative rotation gains. Without being able to properly compare positive rotation gains though, there is limited ground to speculate about the potentials of these effects.

5.3.8 Effect of AC2F Smoothing on Noticeability

One area of the AC2F algorithm that definitely could see improvements is the smoothing component. Throughout Experiment 1, some participants mentioned that they noticed the slightly slippery smoothing that AC2F uses. The slippery effect in this case comes from AC2F smoothing from a rotation gain back towards normal head rotation whenever the user stops moving their head. AC2F's smoothing in this case works naturally when using one's body to rotate as the body elastically bobs slightly in the opposite direction after stopping. On the other hand, this elastic bob does not occur as strongly when the user only rotates their head without using their body. This type of rotation creates a somewhat sliding smoothing effect as participants notice that they keep rotating slightly after they stop their head.

Given that participants mentioned they noticed the smoothing effect at times, it might have had some effect on the estimated detection thresholds. Ideally, the smoothing algorithm should be able to smooth out changes between positive and negative rotation gains while still feeling natural to the user whenever they do smaller head rotations that do not move their body. One way to mitigate the slight slide in rotation that participants may notice when stopping their head rotation is to temporarily increase the speed of smoothing so it is not as noticeable. In the end though, there were not enough resources to implement this within the scope of this thesis.

Given that smoothing components of redirection algorithms likely affect the noticeability of redirection, it presents some more opportunities for future research. It would be interesting to see some research on the noticeability effects of various smoothing approaches and what participants consider as most comfortable. Understanding what works best in terms of smoothing would be very useful for the sake of user comfort if redirected walking ever becomes more common for consumers.

5.3.9 Correlation Analysis Discussion

The primary insight that the correlation matrix points towards is there being some correlation between prior VR experience and the noticeability of redirection. It is hard to draw too much information out of the data, but it appears that the participants reporting the highest amount of VR experience also were among the most sensitive ones. As such, higher VR experience could affect sensitivity towards redirected walking by making it easier to detect redirection. Participants with no prior VR experience could be considered as among the most insensitive. This would be the case if we consider that several participants in this category did not detect any redirection at all. As for those in-between 0 and 6, there is a larger amount of variability in terms of sensitivity to redirection. While studies have shown that prior game experience has not significantly had any effect on noticeability [20], it would be interesting to see future research look deeper into how prior VR experience affects redirection sensitivity.

Thus, this chapter has finished. The following chapter focuses on Experiment 2, which consists of testing the effectiveness of distractors using the developed AC2F redirection algorithm.

6 Experiment 2: Effectiveness of Distractors

This chapter consists of the Method, Results and Discussion components for Experiment 2. The experiment itself focuses on various effectiveness metrics for the developed redirected walking solution.

6.1 Method

As part of developing the extension to the redirected walking toolkit, one of the implemented components was the AC2F algorithm. Since AC2F makes use of more predictive information compared to S2C, it would be interesting to see whether this provides a more effective solution when using distractors. As such, this experiment compares two different conditions. The first makes use of S2C only throughout an entire playthrough of Ensemble Retriever while the other uses S2C during walking and AC2F during the standing distractor battles. The redirection gains employed in this experiment are derived from the estimated thresholds in Chapter 5. This playthrough of Ensemble Retriever is also somewhat shorter than in Experiment 1, taking roughly 15-20 minutes instead of ~30 minutes in hopes that it would make it easier to gather participants. The post-test questionnaire from Experiment 1 was also used for this experiment.

This section consists of the related methods that have been used to measure and test the effectiveness of employing AC2F when using distractors.

6.1.1 Hypotheses

In order to test effectiveness, it is first necessary to operationalise the term. For this experiment, there are two primary types of effectiveness that have been in focus. The first of these is Azmandian et al.'s relative effectiveness [3]. Relative effectiveness provides a measurement of redirection effectiveness through the number of resets that happen throughout one condition compared to another. It is defined with the following formula:

$$RelativeEffectiveness_{Algorithm} = \frac{ResetCount_{ControlCondition} - ResetCount_{ExperimentalCondition}}{ResetCount_{ControlCondition}}$$

This results in the percentage difference of the experimental condition compared to the control counterpart.

The second effectiveness metric that this experiment focuses on is effectiveness in terms of the time in seconds it takes for the user's future path to be aligned to the centre of the room. The distractor solution is in general agnostic to the redirection algorithm that is employed and simply disables any redirection whenever the future virtual path is aligned with the physical centre. With this approach, it is simple to compare S2C and AC2F for this effectiveness metric.

Given the two effectiveness metrics that the experiment focuses on, two null hypotheses have been established:

H_{0_1} : Employing S2C+AC2F in Ensemble Retriever does not provide significantly better relative effectiveness compared to S2C only.

H_{0_2} : Employing S2C+AC2F in Ensemble Retriever does not result in significantly lower time needed to align the user's future path towards the centre of the room compared to S2C only.

6.1.2 Data Collection

Experiment 2 employs a similar data collection procedure to Experiment 1. It records data every frame that Ensemble Retriever is running and focuses on recording data that is relevant to the two established null hypotheses. Some of these include data on when and how many resets that have been triggered, how much time it takes to be aligned with the centre of the room, and the time it takes to defeat a distractor. A full list with descriptions for each piece of recorded data can be found in Section B.3.2.

6.1.3 Data Post Processing

One problem that can affect the results for testing both hypotheses is what could be considered as "unintentional resets". These are resets that can happen during distractor battles in two different ways:

1. If the user is walking fast enough, the buffer between a distractor triggering and a reset triggering might be too small. By the time the user stops after triggering a distractor, they have reached the boundary for triggering a reset as well. This means that the user is almost instantly aligned to the physical centre.
2. If the user moves around a lot during battles they risk triggering a reset since they already are close to the physical bounds while fighting.

Both of these cases could be considered as resets that are unintentional and would skew the results. In terms of frequency, it is the first of these cases that was the most apparent from observation of participants as they played. As such, a data processing step was conducted to attempt removing as many of these unintentional resets as possible.

In terms of removing unintentional resets, there are two potential options. The first of these is to simply not include any resets that happen during battles with distractors. This is the simple solution, but may not end up including legitimate resets that happen between the start and end of a distractor's death animation. A distractor is considered as inactive after this death animation has finished. There are cases where participants start to walk right as a distractor's death animation starts, which means that this method is not ideal as it may miss resets that happen in this time range.

The second option is to discard any resets that happen within a given time buffer from when a distractor spawns. This approach is harder from a post-processing perspective, but should provide more accurate results. This approach can include the legitimate resets that may happen at the end of a distractor's life cycle while minimising the number of unintentional resets. As such, this is the option that was chosen to use for post-processing. In order to decide the length of the time buffer

in this approach, it is first necessary to understand the minimum time a distractor battle has taken. For the recorded data in this experiment, the minimum time was ~ 18.23 seconds. Since this time is recorded at the end of a distractor's death animation, it is necessary to provide some buffer so that legitimate resets can be identified.

Given the minimum distractor duration of 18.28 seconds and the need for some buffer, the post-processing uses a time of 10 seconds. This means that any resets that happened within the first 10 seconds of a distractor spawning become discarded from the analysis. A walkthrough for the entire post-processing procedure can be found in Section [B.3.3](#).

Adding Penalties for Failed Alignments

One element that may differ between algorithms is the rate of failed alignments. Failed alignments are in this case defined as the failure to align the user's future virtual path to the centre of the physical room throughout the lifetime of a distractor. Given that these failure cases may be different between the two conditions, it is necessary to factor them into the hypothesis testing in some way. For this thesis, the problem is handled by providing a time penalty whenever an algorithm fails to align the user properly when testing H_{0_2} . This way, each spawned distractor results in either a successful or unsuccessful alignment time, which can be factored into the data analysis. The penalty itself could technically be any arbitrary value, although it is likely more reasonable to find a value that is grounded into the results of the performed experiment. This way, the skew that the penalty provides is more justifiable than just choosing any arbitrary value. It should be noted that the penalty also needs to be static across all failed alignments to not cause biased results.

The chosen value for the time penalty in this thesis is the average time needed to defeat a distractor, which in this case was 54.86 seconds. The average successful alignment times primarily were within the scope of ~ 23 seconds. As such, the average time taken to defeat a distractor is thus sizeably different compared to the successful alignment times and could provide a reasonable skew in the data if there is a large difference in failure rates. Given how alignments only happen during distractor battles, the average time needed to defeat a distractor is also more closely tied to these measurements. Despite the usage of this penalty, it could also potentially be considered as a bit too generous and not result in a large enough skew to provide significant results. An additional discussion around the use of this penalty in retrospect be found in Section [6.3.1](#).

6.1.4 Changes in Experiment Environment

There was a two week period between Experiment 1 and 2. During this time, one of the ceiling mounted HTC Vive lighthouses disappeared from the experiment setup with no trace or information on where it had gone. As such, it was necessary to set up a new lighthouse in a slightly different configuration. This configuration can be seen in Figure [39](#). The disappearance of the original lighthouse was discovered on the weekend before the experiment started and as such, only limited amounts of room calibration testing was possible. This did create some calibration related problems for two participants. In particular, the reset boundaries became too wide for certain walls and resulted in resets not triggering. These calibration problems were fixed with a software workaround as soon as they were discovered and as such only affected two participants. The number of times



Figure 39: This image shows the alternative lighthouse setup that had to be made as the original ceiling mounted lighthouse disappeared before Experiment 2 started. Instead of a ceiling mounted solution, this setup makes use of a glass mounted lighthouse.

the resets failed to trigger for each of these two were also noted down so they could be considered during post-processing.

6.1.5 Participant Sample

This section consists of information on how the participant sample was acquired, an overview of the demographic in the sample, what information participants were given and miscellaneous information.

Sampling Procedure

The sampling procedure for Experiment 2 was the same as with Experiment 1. It consisted of advertisement by supervisors during lectures, information posters that were placed around campus, advertisement on the local NTNU Discord¹ server and general convenience sampling. It should be noted that participants who already took part in Experiment 1 were still allowed to participate in Experiment 2.

A variety of time slots were assigned to a doodle poll² where participants could sign up with an alias for whatever time slot best fit their schedule. Participants were once again compensated with some chocolate for their time, albeit with less than in Experiment 1 since the amount of time spent for this experiment was shorter. In total, 15 participants signed up for Experiment 2, albeit only 13 of these showed up. These 13 were randomly distributed between the two conditions using the Mersenne Twister library [43], previously mentioned in Chapter 4. This random distribution

¹<https://discordapp.com/>

²<https://doodle.com/>

resulted in 7 participants for the control/S2C only condition and 6 in the experimental/S2C+AC2F condition.

Acquiring participants for this experiment was generally more challenging as many students mentioned they already were fatigued by constantly taking part in the experiments of various master students. As such, a similar sample size to Experiment 1 was not possible to achieve within the same amount of sampling time.

Sample Demographic

The sample itself primarily consisted of students within the age range of 18-24 years. Everyone in the sample identified as male and only one had to remove any optical corrections while participating. 11/13 participants had previous experience with redirected walking experiments due to a larger overlap from participants who already had taken part in Experiment 1. The general spread of VR experience is not quite as well distributed in this sample as compared to Experiment 1. This distribution can be seen in Figure 40 where most participants answered with "1" and "5" on a Likert scale in terms of their prior VR experience.

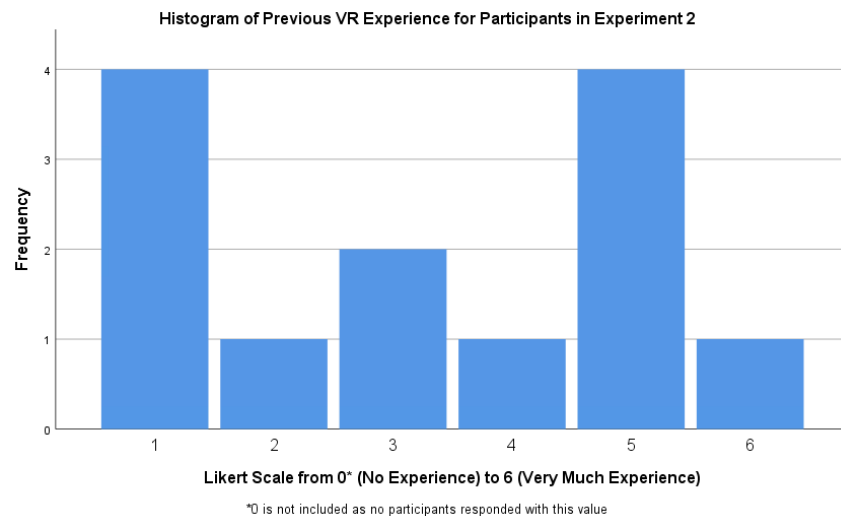


Figure 40: This histogram shows the frequencies of what values participants provided in the demographic questionnaire of experiment 2 in terms of prior VR experience.

Information and Consent

In terms of information and consent, participants were given a sheet that was very similar to the one in Experiment 1. Both information/consent sheets can be seen in Appendix D. Participants were also given some oral information before playing through Ensemble Retriever. This was the same information which was given in Experiment 1, albeit without any mention of detection related information.

The specific details and wording that was used for this oral information can be found in Sec-

tion [B.3.1](#).

Miscellaneous Information and Issues

As mentioned in Section [6.1.4](#), there were some problems related to the room calibration of the new lighthouse setup that had to be done. These problems affected participants 1 and 2 in particular, where participant 1 experienced one failed reset while participant 2 experienced two failed resets. Failed resets in this case are defined as the reset not triggering when it should. In order to avoid potential skewing of data, the number of times that the resets failed to trigger have been added to these participants during the data post-processing step which is mentioned in Section [6.1.3](#) and fully detailed in Section [B.3.3](#).

6.1.6 Changes in Ensemble Retriever Between Experiment 1 and 2

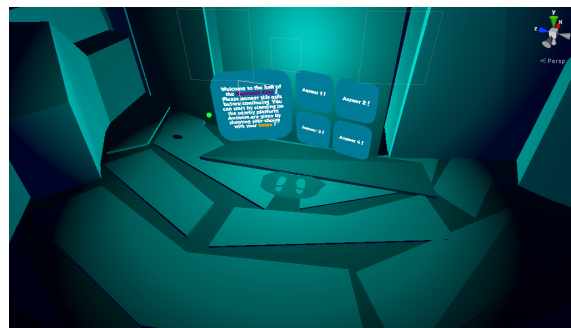


Figure 41: This screenshot from the Unity scene view shows the updated "Hall of The Mountain King" after finishing Experiment 1. Some additional flooring has been added due to participant feedback to make traversal more natural.

A variety of things within Ensemble Retriever were changed between Experiment 1 and 2. The full change-log can be found in the project repository [\[42\]](#) by looking at the commit history, although this section will provide a summary of the changes that were implemented for Experiment 2. Some of these changes only apply when Experiment 2 is chosen as the active experiment in the software. These changes are marked with "(EX2)" while generic changes that apply to the entirety of Ensemble Retriever are marked with "(ALL)". The summary of changes are as follows:

- Overall walking distance has been reduced to allow for a shorter experiment. (EX2)
- The health of the Mountain King has been cut in half to allow for a shorter experiment. (EX2)
- The tutorial does not provide any information on detection of redirection as this is only needed for Experiment 1. (EX2)
- S2C dampening is re-enabled as it no longer can affect the main focus of data collection. (EX2)
- The estimated detection thresholds from Experiment 1 are used as the redirection gains when playing. (EX2)
- An extra plane was added to the floor in the "Hall of The Mountain King". (ALL)

- This was changed due to feedback as participants mentioned it was awkward to walk between the large holes between the rocks in the floor. The difference can be seen by comparing Figure 8 and 41.
- The sound effect volume for absorbing projectiles has been increased to avoid being drowned out by the music. (ALL)
 - This was changed due to participant feedback. Some participants noted that the music during the battle with the Mountain King was a bit loud, so it was hard to clearly hear whenever they managed to absorb any projectiles. Hearing this audio is very useful when not directly looking in the direction the player is holding their hand. As such, the volume of the absorption sound effect was increased.
- The floor in the "Hall of The Mountain King" has been better calibrated as the original calibration was lower than it should be, making some participants notice that their height was slightly inconsistent. (ALL)
- The buffer between reset and distractor triggers has been increased by 0.5m. This increase means that resets trigger 0.5m away from physical walls while distractors trigger at 1.5m instead of the previous 1.0m. (ALL)
 - The reasoning behind this is that there were observations during Experiment 1 where participants would gradually increase their walking speed as they became more comfortable with the experience. Towards the end, walking speeds were fast enough that there would not be a large enough buffer between distractor and reset triggers, making both trigger instead of only the distractor. By increasing the buffer, the effective size of the walking space is slightly reduced, but should decrease the number of situations where resets trigger right after the distractor itself.
- Distractors like the Contrabass, Oboe and certain phases of the "Mountain King" have increased projectile speeds. (ALL)
 - This was based on participant feedback for certain projectile attacks taking too long to reach the player, and thus being frustratingly slow.
- The hint providing fireflies in the environment now change colours after the player has visited them. (ALL)
 - This was implemented due to participant feedback as a means to make it easier for somewhat disoriented players to see which of the fireflies they already had visited.
- The Distractor trigger magnitude cooldown has been decreased from 1.75m to 1.5m. (ALL)
 - This was primarily done due to observations of situations where participants would walk a larger distance right as the death animation of a distractor starts. By the time it was finished and the cooldown initiated, the participant was almost at the bounds of the physical space, resulting in a reset instead of a distractor trigger.

6.2 Results

This part of the chapter provides the results of the experiment related to the two established hypotheses as well as insights gained from demographical information. Like with Experiment 1, data visualisation is provided by SPSS due to the large data file sizes. A significance level of 0.05 was used for all statistical tests. For the sake of disclosure for any graphs that make use of participant ID's, the author's ID is 5. Just like in Experiment 1, this section only provides the results of this experiment. The discussion around these results can be found in Section 6.3.

6.2.1 Relative Effectiveness

The first of the null hypotheses to be tested is effectiveness in terms of relative effectiveness.

Minimum Time Needed to Defeat Distractors

As mentioned in Section 6.1.3, in order to find a reasonable period to discard unintentional resets in, it is first necessary to look at the minimum time needed to defeat any distractor. This information can be seen in Figure 42 where the minimum time is ~18.23 seconds.

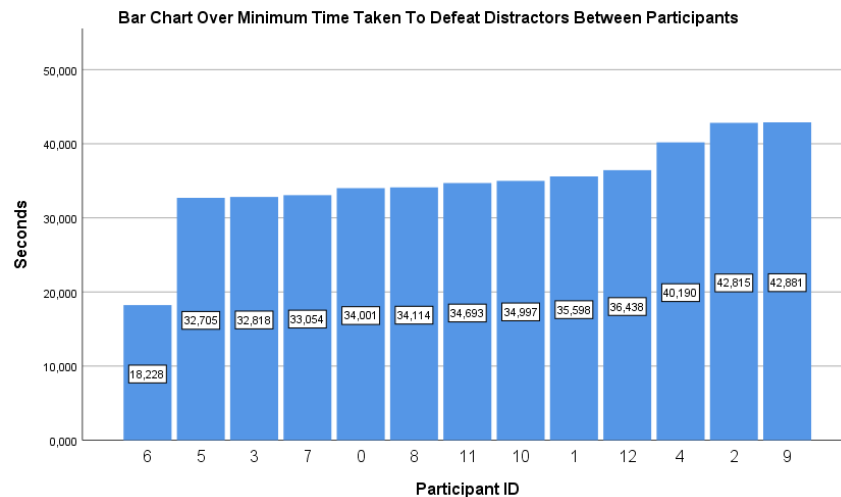


Figure 42: This bar chart shows the minimum time needed for each participant to defeat any distractor in Ensemble Retriever.

Time and Walking Speed Normalisation

Two variables that potentially could have some effect on the reset counts between conditions is the time spent on walking and the walking speed of participants. It is thus necessary to first look at the mean movement speed and mean time spent on walking between the conditions before further analysis can take place. A boxplot on time spent walking for participants between the two conditions can be seen in Figure 43. A boxplot on the mean metres per second that participants walked at between conditions can be seen in Figure 44.

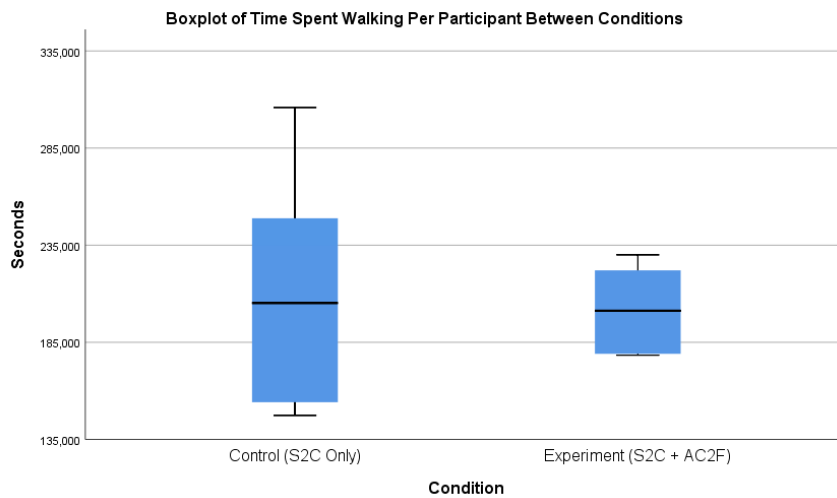


Figure 43: This boxplot shows the time that participants spent on walking between the two conditions.

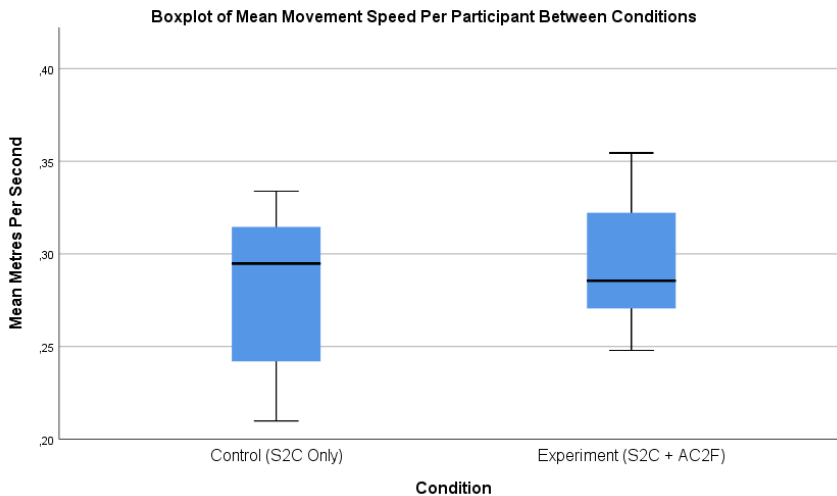


Figure 44: This boxplot shows the mean walking speeds of participants between conditions.

A Shapiro-Wilk test was employed to check the normal distribution of these two variables. For the time spent walking, the test provided $p = 0.195$ for S2C Only and $p = 0.490$ for S2C+AC2F. Both of these conditions are as such normally distributed, and an independent samples t-test was employed to look for statistically significant differences. The independent samples t-test yielded $t = 0.262$, $p = 0.798$, which means there is no statistical significance in terms of time spent walking. For clarity's sake, the mean walking time for S2C Only was 209.2 seconds and 202.0 seconds for S2C+AC2F.

In terms of the mean walking speed, the Shapiro-Wilk test provided $p = 0.472$ for S2C Only and $p = 0.847$ for S2C+AC2F, meaning that the data is normally distributed. As such, an independent samples t-test was employed to look for statistically significant differences. The independent samples t-test provided $t = -0.633$, $p = 0.540$ showing that there is no statistically significant difference for walking speeds either. For clarity's sake, the mean walking speed for S2C Only was 0.278 metres per second and 0.294 metres per second for S2C+AC2F.

Since there is no statistically significant difference between the two conditions in terms of time spent walking or walking speed, further analysis will focus on the mean number of legitimate resets per participant. If there had been a significant difference, a time or speed normalised variable would have needed to be calculated for comparisons.

Number of Resets for Participants Between Conditions

The mean number of resets that participants experienced between conditions can be seen in Figure 45. To supplement the means, a boxplot of the same data can be seen in Figure 46. As a reminder, the data sample used for these graphs consists of legitimate resets only. Unintentional resets have been removed using the data post-processing step as mentioned in Section 6.1.3.

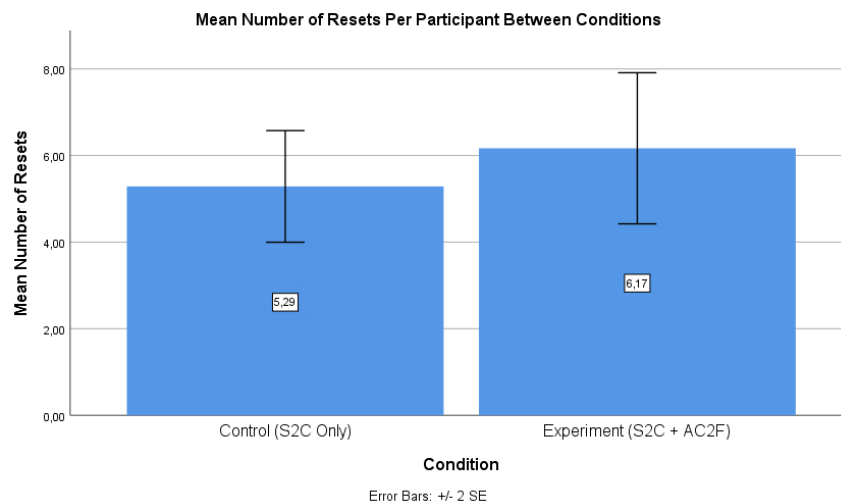


Figure 45: This bar chart shows the mean number of resets that participants experienced between conditions.

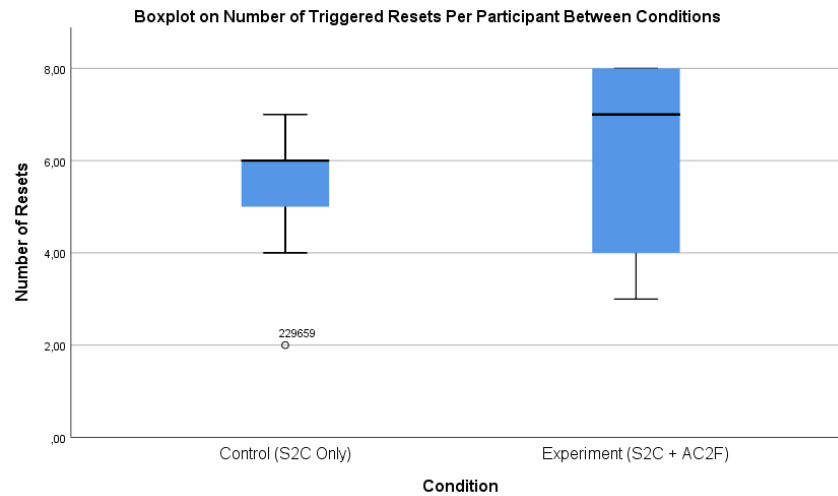


Figure 46: This boxplot shows the spread of resets counts that participants experienced between conditions.

Hypothesis Testing

Finally, in order to test H_{0_1} , it is first necessary to test the normality of the data. Using a Shapiro-Wilk test, the S2C Only condition yielded $p = 0.036$ while S2C+AC2F yielded $p = 0.096$. As such, the S2C Only condition is not normally distributed and a non-parametric significance test is needed. In this case, the Mann-Whitney U non-parametric significance test is used.

Looking at the boxplot in Figure 46, the overall shape of the distribution is fairly different between the two conditions. As a result, the comparison in this case will be on mean ranks only. The Mann-Whitney U test results in $U = 12.500$, $p = 0.234$. From this, we can conclude that there is no statistically significant difference in terms of the number of resets between the two conditions and H_{0_1} is supported.

Since there is no significant difference, the relative effectiveness between the two conditions has not been calculated.

6.2.2 Alignment Time Effectiveness

The second null hypothesis to test is related to effectiveness in terms of time needed to align the user's future path towards the centre of the physical space.

Alignment Failure Rates

Before looking at the processed data sample, it is first necessary to understand the differences in alignment failure rates between the two conditions. Participants in the S2C Only condition triggered 92 distractors in which 22 failed alignment. This ratio results in a failure rate of 23.9%. Participants in the S2C+AC2F condition triggered 74 distractors in total and consisted of 6 failed alignments. This ratio results in a failure rate of 8.1%. The time taken to defeat the distractor during these failed cases can be seen in Figure 47 while the level of the participants' conducting baton during the failure

can be seen in Figure 48. As seen in Figure 48, the S2C Only condition did fail in some cases where the participants' baton was at a lower level, meaning that some failures happened during longer battles. The S2C+AC2F condition on the other hand, only failed when the participants' baton was at the maximum level, meaning the alignments failed during the shortest battles.

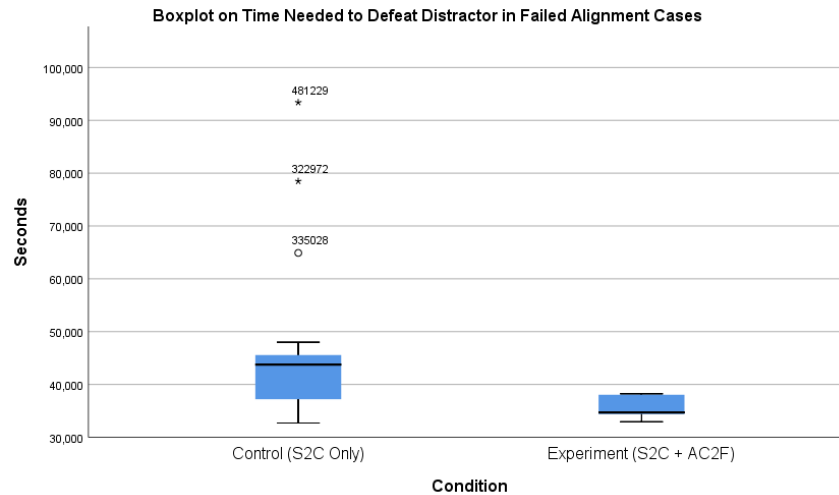


Figure 47: This boxplot shows the spread of time in seconds needed to defeat a distractor during the times where alignment towards the centre of the room failed.

Time Needed for Alignment Between Conditions

Finally, the boxplot showcasing the spread of alignment times with failure penalties can be seen in Figure 49. The mean time taken to align the user towards the centre in this case is 30.91 seconds for S2C Only and 25.34 seconds for S2C+AC2F. As a supplement, the spread of alignment times for only successful cases can be seen in Figure 50. If failure penalties are disregarded, the mean time needed for alignment is 22.55 seconds for S2C Only and 22.85 for S2C+AC2F.

Hypothesis Testing

In order to test H_{0_2} , the data sample including failure penalties will be used. A Shapiro-Wilk normality test was first conducted on the alignment time data. This test provided $p < 0.001$ for both conditions and as such, neither is normally distributed. As such, the non-parametric Mann-Whitney U test is used for significance testing. The comparison is made with mean ranks as the shapes in Figure 49 are fairly different. This comparison results in $U = 2948.500$, $p = 0.276$, meaning that the test cannot find any statistically significant difference in alignment times. This result means that the H_{0_2} null hypothesis is supported. Despite this, it should be noted that the alignment fail rates for the two conditions are different and that the S2C+AC2F condition consisted of 15.8% more successful alignments compared to S2C Only.

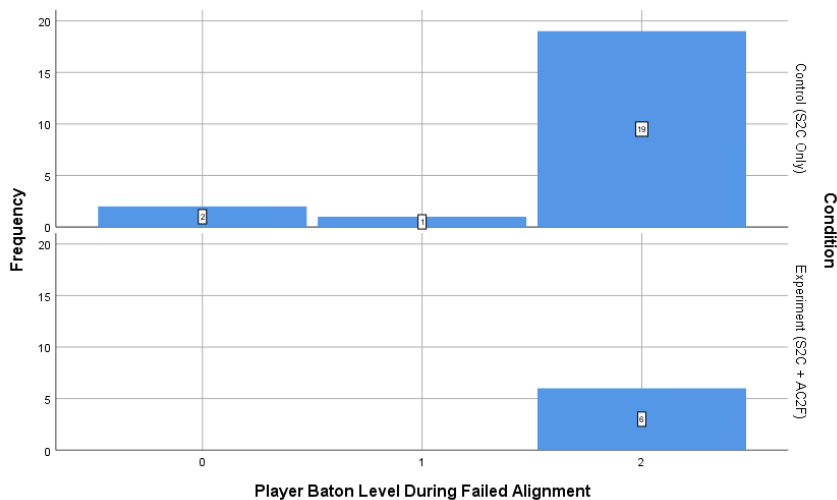


Figure 48: This histogram shows the frequencies of which level the participants’ baton was at during failed alignment. Higher levels allows the player to deal more damage.

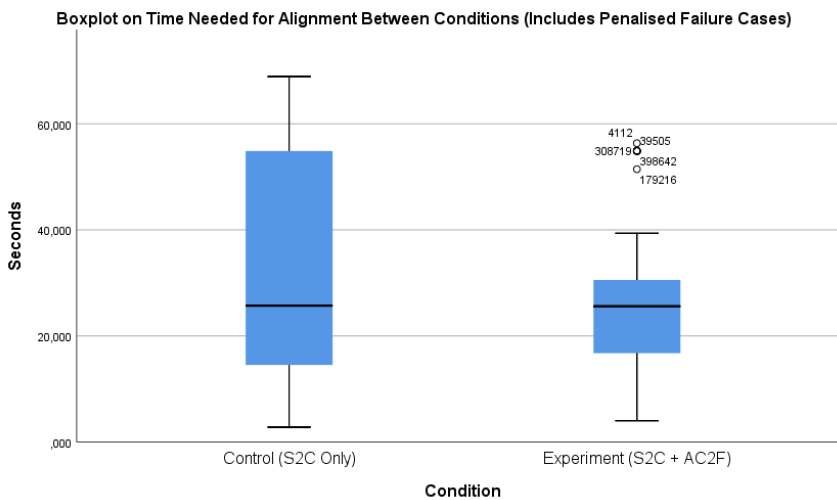


Figure 49: This boxplot shows the spread of time needed to align participants towards the centre of the room. It includes the penalties that are given to cases of failed alignments.

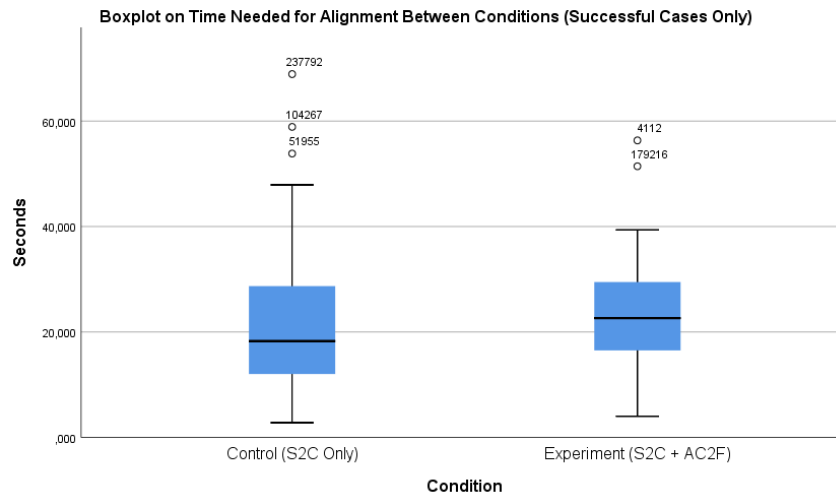


Figure 50: This boxplot shows the spread of time needed to align participants towards the centre of the room. In this case the data consists of only successful alignments.

6.2.3 Supplementary Graphs: Head Rotation for Participant 5

To provide some supplementary illustration for a discussion in Section 6.3.2, a graph showing the changes in head rotation for the author while taking part in Experiment 2 can be seen in Figure 51. Since large fluctuations in frame latency between walking and distractor battles could affect Figure 51, an additional graph showing the frame latencies throughout the experiment can be seen in Figure 52.

6.2.4 Supplementary Graph: Time Taken to Defeat Distractors and Player Baton Level

To provide some supplementary information for a discussion in Section 6.3.4, a graph showing the time needed to defeat distractors in this experiment can be seen in Figure 53. Furthermore, the mean time taken to defeat distractors when the participants' baton level was at the highest level was 44.21 seconds.

6.2.5 Supplementary Graph: Mean Walking Speed and Prior VR Experience

To illustrate one correlation that is discussed in Section 6.3.6, a boxplot on the mean walking speeds of participants concerning prior VR experience can be seen in Figure 54.

6.2.6 Demographic Insights

Two correlation matrices have been generated from the demographical data in relation to the primary data that is relevant for the hypotheses that have been tested. The Spearman's Rho correlation test has been used to generate the correlation matrices as some of the data is not normally distributed. These correlation matrices can be seen in Figure 55 and 56. The correlation test is split into two matrices, primarily due to differences in data structure between some of the variables.

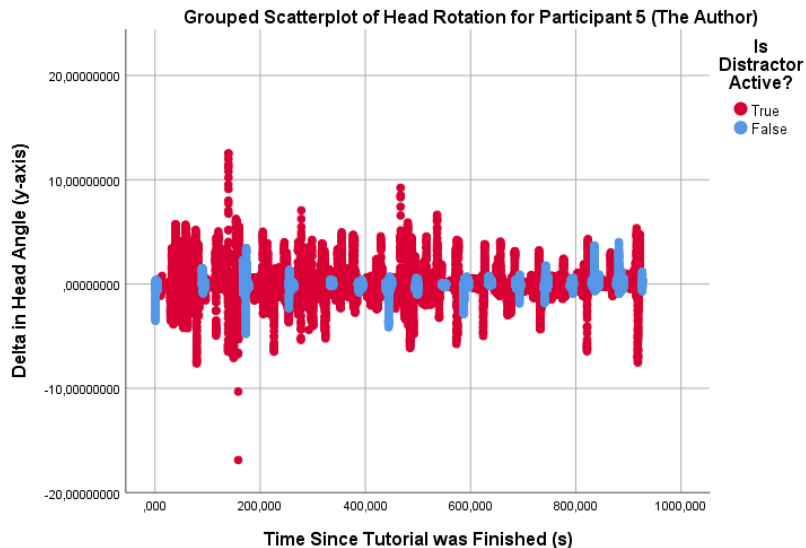


Figure 51: This scatterplot shows the deltas in head rotation for the author while participating in Experiment 2.

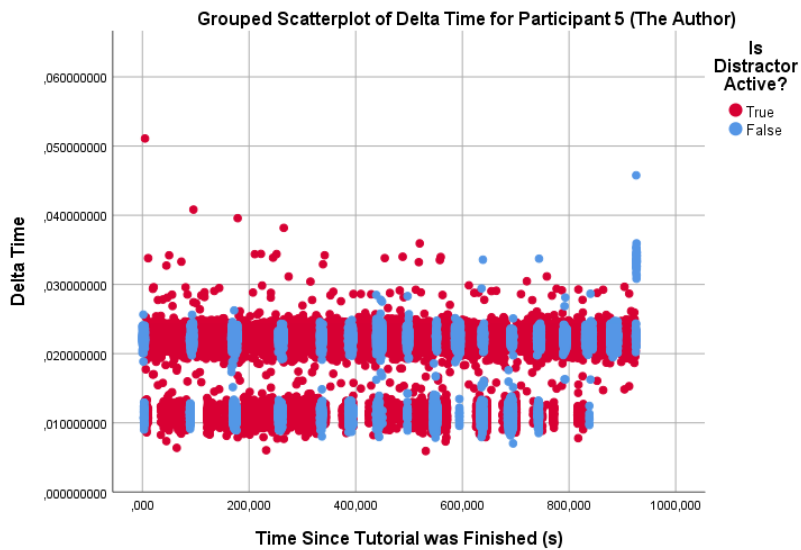


Figure 52: This scatterplot shows delta time/frame latency throughout Experiment 2 for participant 5.

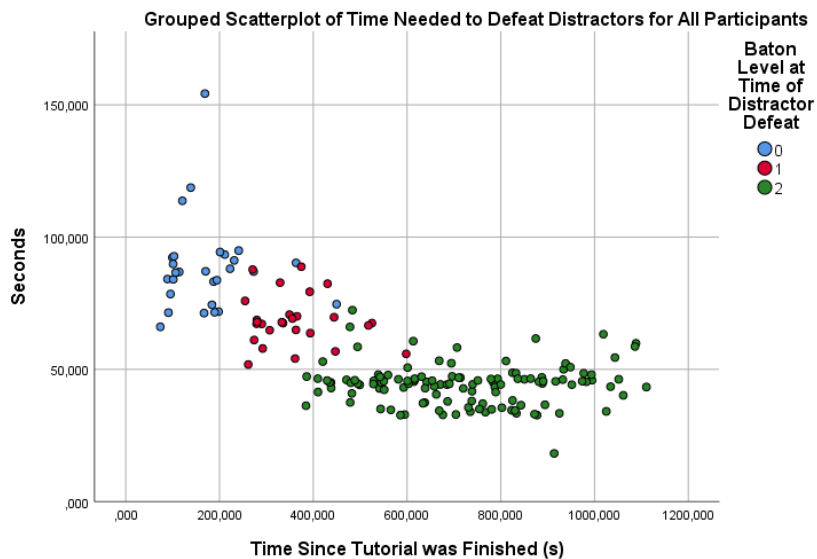


Figure 53: This scatterplot shows the time needed to defeat distractors among all participants throughout Experiment 2. It is grouped by baton level as higher levels result in higher damage and shorter battles.

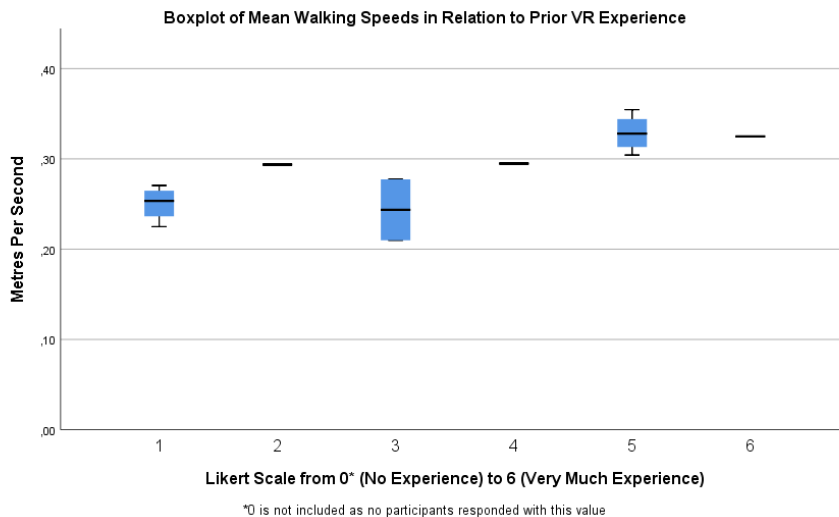


Figure 54: This boxplot shows the mean walking speeds of participants from Experiment 2 in relation to their reported prior experience with VR.

Correlations

Spearman's rho	Age Range	Age Range	Need to Remove Optical Corrections	Prior RDW Experience	Prior VR Experience	Time Spent Walking	Mean Walking Speed	Number of Legitimate Resets	Number of Failed Alignments
	Correlation Coefficient	1,000	,157	,232	-,161	,223	,000	,429	-,556*
	Sig. (2-tailed)		,609	,446	,599	,463	1,000	,144	,049
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	,157	1,000	,123	-,278	,309	-,386	,039	-,245
	Sig. (2-tailed)	,609		,689	,357	,305	,193	,898	,420
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	,232	,123	1,000	-,528	,171	-,342	,350	-,332
	Sig. (2-tailed)	,446	,689		,063	,577	,253	,242	,268
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	-,161	-,278	-,528	1,000	-,693**	,835**	-,271	,226
	Sig. (2-tailed)	,599	,357	,063		,009	,000	,371	,458
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	,223	,309	,171	-,693**	1,000	-,775**	,020	-,090
	Sig. (2-tailed)	,463	,305	,577	,009		,002	,949	,770
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	,000	-,386	-,342	,835**	-,775**	1,000	,006	-,038
	Sig. (2-tailed)	1,000	,193	,253	,000	,002		,985	,902
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	,429	,039	,350	-,271	,020	,006	1,000	-,767**
	Sig. (2-tailed)	,144	,898	,242	,371	,949	,985		,002
	N	13	13	13	13	13	13	13	13
	Correlation Coefficient	-,556*	-,245	-,332	,226	-,090	-,038	-,767**	1,000
	Sig. (2-tailed)	,049	,420	,268	,458	,770	,902	,002	
	N	13	13	13	13	13	13	13	13

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 55: Correlation matrix between various demographical variables and relevant variables in Experiment 2.

			Age Range	Need to Remove Optical Corrections	Prior RDW Experience	Prior VR Experience	Time Taken Until Alignment
Spearman's rho	Time Taken Until Alignment	Correlation ...	-.291**	-.055	-.068	-.001	1,000
		Sig. (2-tailed)	.000	.490	.392	.989	.
		N	162	162	162	162	162

** Correlation is significant at the 0.01 level (2-tailed).

Figure 56: Correlation matrix between the time needed to align participants to the centre of the physical space and various demographical variables in Experiment 2.

Qualitative Feedback

In terms of qualitative feedback, this section provides a summary of the feedback that participants provided in the questionnaire. The summary is structured after the questions that did receive answers.

Whether the Participants Found Any Bugs or Glitches

In general, some participants noted that they experienced a loss of controller tracking at times. Other than that, no major bugs were experienced by participants.

Whether the Participants Found the Experience Enjoyable

In general, the participants found the experience to be very enjoyable, although one noted that it was too long for their liking.

How the Participants Felt About the Redirection Techniques

Some participants did not notice any redirection, while others did notice it at certain times. The general feedback from participants that did notice the redirection at times was that it felt natural as it was not too strong. In terms of when participants noticed the redirection, it varied a bit. Some had an easier time noticing it in battle, while others found it more easy to notice while walking. The distribution of participants mentioning they noticed the redirection was equal between the two conditions (3 in S2C Only and 3 in S2C+AC2F).

Whether the Participants Had Any Problems Throughout Their Experience

The main feedback that was provided for this question was that the physical cable that the HTC Vive uses was rather frustrating to deal with. There were times where participants felt like they had gotten tangled up in the cable and needed to spent some time untangling themselves, somewhat breaking the immersion of the experience. Participants mentioned that the experience would have been substantially improved if integrated with wireless HMD's in the future.

6.2.7 Summary

The following paragraphs will summarise the results that have been presented in relation to their relevant hypotheses.

H_{0_1}

To summarise, the first null hypothesis was: "Employing S2C+AC2F in Ensemble Retriever does not provide significantly better relative effectiveness compared to S2C only". There was no statistically significant difference between the two conditions in terms of the mean number of resets per participant (Mann-Whitney U: $U = 12.500, p = 0.234$). As such, no relative effectiveness was calculated and the H_{0_1} null hypothesis is supported.

H_{0_2}

To summarise, the second null hypothesis was: "Employing S2C+AC2F in Ensemble Retriever does not result in significantly lower time needed to align the user's future path towards the centre of the room compared to S2C only". There was no statistically significant difference between the two conditions in terms of mean time taken to align participants to the centre of the room (Mann-Whitney U: $U = 2948.500, p = 0.276$). Despite this, it should be noted that the alignment failure rates between the two conditions were different. Using S2C together with the AC2F algorithm resulted in 15.8% fewer alignment failures compared to using only S2C. While the H_{0_2} null hypothesis is supported, the difference in alignment failure rates should be considered.

6.3 Discussion

This section focuses on discussing the presented results in Experiment 2.

6.3.1 Limitations of Employed Alignment Failure Time Penalty

As part of testing H_{0_2} , no statistically significant difference was found in terms of mean alignment times between the two conditions. Despite the result not being significant, there was a 15.8% difference in terms of alignment failure rates between the two conditions. Given that this percentage difference is not reflected in the results of the statistical test, it may be that the chosen alignment failure time penalty was too generous. If we take a look at Figure 47, we can see that most failed alignment cases consisted of distractor defeats, which were slightly lower than the chosen penalty. The primary issue here is the three outliers in the control condition, which were above the time penalty value. In the case of these outliers, their unsuccessful alignment times would get skewed downwards by the penalty rather than upwards as intended. As such, using the mean time taken to defeat a distractor might have been too generous to provide a statistically significant skew in the results.

It is, of course, possible to choose a more substantial penalty to attempt providing more significant results, but this would likely be too arbitrary. Another option would be to use the maximum time taken to defeat a distractor rather than the mean time. This choice would result in a far harsher penalty, but it would still be grounded in the results of the experiment itself. In the end, it may be best to primarily focus on the percentage difference in failure ratios, rather than the results from the statistical test for this hypothesis. This percentage difference is not affected by the potential limitations of a too generous time penalty and would as such, be a better focus for discussion for the rest of this section.

6.3.2 Relationships Between Alignment Fail Rates and Mean Number of Resets

One of the most interesting results from Experiment 2 is that there was no significant difference in terms of the mean number of resets, but there was a 15.8% difference in failure rates. Given the higher fail rate for using S2C only, it would be expected that this condition in general results in more resets as it fails to align the user to the centre. The question then becomes: why is this not the case? It may be that while only using S2C results in more failures to fully align the user towards the centre, it may be close enough to avoid them moving into the reset bounds of nearby walls. Of course, the results may also be like they are due to the sample size, but thinking about the current results provides some interesting ideas.

”Orbital Scenarios”

While this is just speculation, if the user is at the corner of a room and starts to walk parallel to a wall rather than towards the centre, it may result in longer curved paths. It should be noted that this effect only applies to situations where longer straight paths are taken. This effect is illustrated in Figure 57 in what we could consider an ”orbital scenario”. The illustration relies on curvature gains only as the amount of head rotation while walking is generally limited. To show an example of the differences in head rotation between walking and battle states, see Figure 51 and the supplementary Figure 52. The first of these shows the changes in head rotation throughout the experiment while the author participated, and the second figure shows the related frame latencies between the two states. The frame latencies between the two states appear to be relatively similar and as such, should not have an impact on Figure 51.

The effectiveness of the aforementioned orbital scenarios would likely vary depending on where in the room the user is after distraction is finished. The top right case in Figure 57 consists of the user starting in a corner space and starting to walk roughly in parallel with any walls. When this happens, they will likely go a further distance until the next distractor triggers compared to the top left case in Figure 57, where there is a straight path towards the centre. The reasoning for this is that S2C as an algorithm will not attempt to apply any curvature gains while moving towards the room centre. As such, the amount of effective redirection is roughly halved as redirection is not applied until the user has moved past the centre again. In another scenario, the user starts at the middle point of a wall and moves roughly in parallel with it as seen in the bottom right example in Figure 57. When this happens, they will likely hit a distractor or reset trigger quite a bit earlier compared to if they started to move towards the centre of the room as seen in the bottom left example in Figure 57.

Due to this potentially happening, it could explain why the failure rate is different while the mean number of resets between conditions is not. If using S2C only results in optimal orbital scenarios at times, while being slightly more inefficient otherwise, the total effectiveness for using S2C only could average out and become similar to S2C+AC2F.

Optimising the S2C Algorithm During Straight Walking When Using S2C+AC2F

Given the possibility of orbital scenarios, the overall solution of using distractors could be somewhat optimised. It should be noted that this optimisation would work assuming S2C is employed

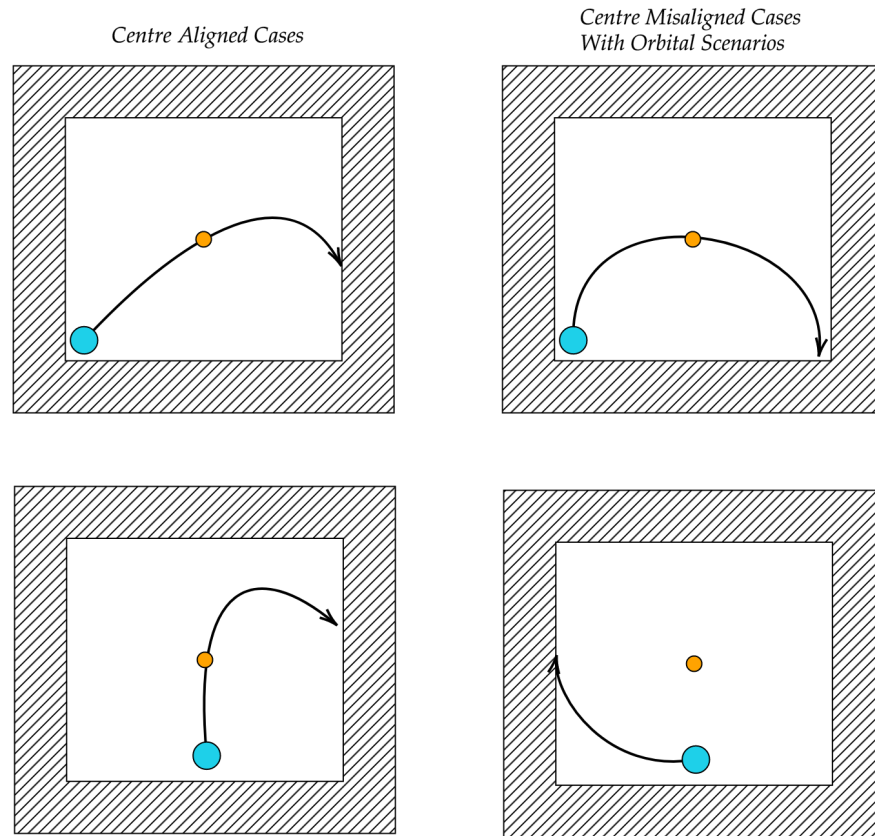


Figure 57: This illustration shows some expected paths that the user can take when using the S2C Algorithm after finishing distractor interaction. For the sake of these illustrations, only curvature gain is considered as the user is not expected to move their head much when walking towards a goal. The illustration also focuses on virtual straight walking situations specifically. Failure to align the user to centre during distraction can result in "orbital scenarios". The effectiveness of these will vary depending on where in the room the user is before starting to walk again.

during walking and AC2F is used with distractors. In cases where users will walk in straight lines, it could be possible to dynamically create the alignment heuristic depending on where they are physically located instead of always aligning to the centre. By optimising the alignment heuristic that AC2F aims to align towards, S2C can provide a more effective curved path after distractor interaction is finished. To some extent, this would result in similar movement to the Steer To Orbit (S2O) algorithm whenever this is optimal. Given that S2O primarily outperforms S2C for physical spaces between $16m^2$ and $31m^2$ [3], it may be best to primarily work with S2C due to its better performance on a general level.

Furthermore, it is crucial to keep in mind that this optimisation would primarily apply to straight walking scenarios when using S2C+AC2F. If straight walking is not expected in the developed experience, it may both be safer and more efficient to simply always set the AC2F heuristic to align to centre. It should be noted that straight paths are one scenario where S2O could outperform S2C though [10]. As such, another option would be to dynamically switch between S2C and S2O when walking depending on the expected walking patterns. This approach would likely require some individual testing on a case by case basis to determine how straight paths are detected. Given that this is implemented, it could be possible to dynamically switch between the S2C and S2O algorithms when walking depending on whether straight paths are expected or not.

Mitigating Failure Cases for the AC2F Algorithm

Another element that can be further optimised is to mitigate the failure cases for the AC2F algorithm. The results show that AC2F only fails during the shortest distractor battles as the player's baton at those points is at the highest damage level. Furthermore, the main reasoning behind these failures is insufficient head rotation, which can only happen if the distractor chooses small random angles to move in. AC2F as an algorithm will always succeed alignment given enough head rotation for any specific situation. S2C on the other hand will not necessarily do this if used in place of AC2F as it does not consider the alignment heuristic that AC2F does. It could be possible to mitigate AC2F's failure cases by introducing a minimum angle for distractor movement rather than using pure randomness. By doing so, we can guarantee that a certain amount of head movement will happen and as such, make sure that the failure rate decreases for the shortest battles. This is of course a rather specific optimisation to the usage of one type of concrete distractors. Despite this, it would be a great benefit whenever a virtual experience makes use of concrete distractors that rely on movement to facilitate head rotation.

6.3.3 Potential Improvements to the Data Processing Approach

Another side of Experiment 2 that could be substantially improved is to streamline and improve the data processing approach. Currently, there is a large amount of post-processing needed to get relevant and usable data which technically could have been recorded more easily and automatically on the software side of Ensemble Retriever. Furthermore, the processing of unintentional resets could also be improved. Instead of cutting off all resets that happen within the first 10 seconds of a distractor spawn, it might be more accurate to instead only *include* resets that happen within the last 10 seconds of a distractor's life. This would likely be a better approach to removing type

2 unintentional resets which were mentioned in Section 6.1.3. As a reminder, type 2 unintentional resets consist of users triggering a reset during distractor battles by moving around as they fight.

In the current approach it is possible to for example have a distractor last for 50 seconds and have a type 2 unintentional reset at 25 seconds due to large amounts of engaged movement. With the current approach, this reset would have been recorded as legitimate instead of unintentional. Fixing this in post-processing would be rather challenging due to the time-serial data format, but doing so on the software side of the recording should be rather feasible.

The need to perform this post-processing in the first place came from observing human factors and behaviour that were attempted to be mitigated from Experiment 1. In particular, mitigating type 1 unintentional resets was the main reasoning behind increasing the buffer between reset and distractor trigger bounds. While this change in general likely mitigated a good amount of unintentional resets, it did not entirely stop them from happening. It would of course be possible to further increase the buffer, but this would risk having the actual walkable space become smaller and more annoying for participants as distractors would trigger with higher frequency. Providing more accuracy in terms of being able to remove unintentional resets would likely help with providing more valid and accurate results.

Another element of the post-processing step that could be improved would be the choice for the failed alignment time penalty as already mentioned in Section 6.3.1. Choosing a harsher penalty could result in more statistically significant results when testing H_{0_2} .

6.3.4 Minimising the Risk of Cybersickness Buildup With Distractors

The results from Experiment 2 showed that the mean time needed for alignment in successful cases for AC2F was ~ 23 seconds. At the same time, the mean time needed to defeat a distractor at the highest baton level was ~ 42 seconds. This provides an interesting time buffer which could be used in a variety of ways. One possibility would be to decrease the employed rotation gains so they better fit the length of a distractors life. This approach would likely contribute to decreasing the risk of hitting various individuals' cybersickness thresholds. Another option would be to keep the current gains and instead decrease the time needed to interact with distractors.

Decreasing the strength of rotation gains would result in longer periods of exposure to redirection. In comparison, if the gains are strong, then alignment will happen earlier and disable redirection until the distractor is finished. Which of these two approaches is ideal in terms of cybersickness is hard to say, but it would have been interesting to see future research look into it. Another element that these two approaches could affect is the adaptation towards positive rotation gains which were experienced in Experiment 1. If there are little to no periods of time where 0 gains are applied to the user, then this might result in different adaptation effects compared to what was experienced in Experiment 1. This potential phenomenon is also something that would be interesting to see further research on.

One thing that should be noted though, is that lowering the gains will likely have a negative effect on the alignment failures that are experienced in shorter battles. If the improvements discussed in Section 6.3.2 were to be implemented, then this would likely make it easier to decrease

gains without impacting alignment failure rates to the same degree. In general though, it would be recommended to still have some buffer between the time needed to defeat distractors and the time needed for alignment in case of unforeseen circumstances.

From a game design point of view, the results show that the majority of the gameplay time is spent on battles. In cases where more exploration is necessary, it might be better for the players to actually spend less time fighting distractors and more time walking around. From this perspective, stronger gains with shorter battles are preferable.

6.3.5 Success of Employing Mean Detection Threshold Gains From Experiment 1

The employed redirection gains in Experiment 2 were derived from mean detection thresholds in Experiment 1. Despite these thresholds being aggregates from rather diverse individuals, the results seem to point towards this approach having been relatively successful. In terms of qualitative feedback, 6 out of 13 mentioned they had noticed they were being redirected at some points during the experience. Despite this, no participants mentioned feeling any cybersickness or nausea as a result from the redirection. As such, it would not appear that the cybersickness thresholds for the individual participants in the sample were exceeded. Employing the same gains for a different population might not have worked quite as well though, as the mean detection thresholds could vary and so could the cybersickness thresholds. The approach of using detection thresholds for redirection gains to strike a balance between effectiveness and comfort will likely only be safe if the gains are derived from estimated thresholds of the target population. Furthermore, since there were no direct cybersickness tests in this experiment, it is hard to fully conclude the effectiveness of the approach other than that uncomfortable levels were not reached.

6.3.6 Correlation Matrix Discussion

This section will further analyse the significant correlations that were found in Figure 55 and 56.

Correlations for Age Range

There are a few significant negative correlations for the age range variable ($r_s = -0.556, p = 0.049$ for number of failed alignments and $r_s = -0.291, p < 0.001$ for time taken until alignment). There is little information that can be gained from these as the vast majority of the participant sample was within the age range of 18-24.

Correlations for Prior VR Experience

In terms of prior VR experience, there appears to be a negative correlation in terms of time spent walking ($r_s = -0.693, p = 0.009$). This correlation is reasonable as one area where this variable has a positive correlation is on the mean walking speed of participants ($r_s = 0.835, p < 0.001$). If we take a look at Figure 54, there appears to be a general trend on walking speeds increasing as higher amounts of prior VR experience is reported. The distribution for this sample is rather skewed towards those reporting a value of 1 and 5 though, so drawing any major conclusions on this hard. It would not necessarily be surprising that higher amounts of experience relate to participants that are comfortable with higher movement speeds in VR though.

Correlations for Time Spent Walking

One negative correlation is between time spent walking and the mean movement speed ($r_s = -0.775, p = 0.002$). This correlation should not be particularly surprising as the time spent walking decreases when the movement speed increases. This correlation is also likely why the mean time spent walking in Figure 43 shows that S2C Only spends a few more seconds as the mean walking speed for the condition seen in Figure 44 is slightly lower. It should be noted though that the recording of time spent walking and mean walking speed is not fully accurate as it simply records whenever distractors are not active. This limitation may result in a small skew as it can include situations where participants stand still and look around for a little bit.

Correlations for Number of Legitimate Resets

The number of legitimate resets is negatively correlated with the number of failed alignments ($r_s = -0.767, p = 0.002$). The number of failed alignments as a variable primarily relates to the level of the participants' baton as well as the amount of random movement that a distractor does. Given the speculation on orbital scenarios in Section 6.3.2, it may explain why in this case failed alignments correlate with a lower amount of legitimate resets.

This concludes the discussion around the results of Experiment 2. The following Chapter 7 consists of a more general discussion for the thesis as a whole.

7 General Discussion

This chapter consists of a more general discussion around the research that has been conducted as a whole.

7.1 Limitations of the Current AC2F Future Path Heuristic

The developed AC2F algorithm relied on sampling the last second of positional changes to determine its future path heuristic. This has worked relatively well for Ensemble Retriever given that AC2F in combination with S2C resulted in 15.8% fewer alignment failures compared to a pure S2C solution. Despite this result, it is worth considering the walking scenario that the algorithms have been used for. For Ensemble Retriever, the explorable space is large and open. This results in reasonably straight paths between fireflies until the player reaches the final portal. Furthermore, the player does not stop at any point and move in the opposite direction, which is favourable for the sampled future path heuristic. If the player on the other hand reaches a point and then decide to do a 180-degree virtual turn, the employed heuristic may not be as effective if they recently interacted with a distractor. In this case, a new distractor would need to be triggered, or a reset would occur.

On the flip side, AC2F as an algorithm does not need to use a sampled future path heuristic. Any sort of directional heuristic could be used. As such, in more confined spaces, it could be possible to use more predictive heuristics for AC2F instead of a sampled future path. In general, if there are situations where the developer knows the direction a user will take, it may be more effective to have some set heuristics which can be used. As far as a generic solution is concerned though, creating a heuristic out of sampled position changes is relatively simple and works well for open areas.

7.2 Salience and Distraction

One element which is worth some further discussion is the pre-existing salience in the virtual environment. As priorly mentioned in Chapter 2, current research has suggested that a larger distribution of salient elements in a scene decreases the effectiveness and focus on other salient objects [33]. In general, the various salient elements in Ensemble Retriever could be summarised as such:

- Environmental glowing mushrooms which always are visible.
- Hint providing fireflies which only are visible outside of battles.
- The portal that sends the player to the Hall of The Mountain King.
- Any distractors that the player battles.
- The projectiles that a distractor fires at the player.

The salience provided by the mushrooms and portal could for example interfere with the salience

that a distractor provides by itself and together with its projectiles. Some participants have also mentioned that they used the mushrooms as a reference point to detect redirection as it made detection easier. This effect could be a result of the salient mushrooms providing a more considerable degree of optical flow in an otherwise darker environment. This effect would also correspond somewhat with the speculation for as to why it was somewhat easier to notice negative rotation gains during battles as the distractors would create additional optical flow as well. One option to mitigate this would be to decrease the overall salience in the scene. Despite this, if we think practically and realistically, a darker scene or environment will usually consist of a certain amount of salient regions for the sake of illumination. As such, this should likely be kept in mind when working with such scenes. In any case, it would be interesting to see further research into how salience could affect optical flow and how this further could affect the noticeability of redirection.

7.3 The Ideal Timing for Switching Redirection Algorithms

Throughout both experiments, one observation of note was when participants started to move again after finishing a distractor battle. Most participants waited until the fireflies became visible again, meaning that the transition from AC2F to S2C was finished. Some participants on the other hand started to move as a distractor was still playing through its death animation. In the current solution, the switch from AC2F to S2C happens when this animation is finished. This means that if the player starts to move as this animation is playing, they will not be exposed to any curvature gains for a little while. Rotation gains would not be applied either if AC2F finished alignment. This means that the effective redirection is decreased for a small space of time.

Another problem with starting to walk this early is that distractors have a distance cooldown that needs to be travelled before they can be spawned again. This cooldown only starts to tick down from movement after the switch from AC2F to S2C is finished. As a result, starting to move early might result in hitting a reset instead of a distractor as it still is on cooldown. This situation can happen as the cooldown did not track how much the player moved before the algorithm switch. One option to deal with this would be to switch algorithms as soon as a distractor's health reaches zero, but this also has a downside. If AC2F did not finish alignment, then this small remaining time could be used to apply further redirection and potentially complete alignment.

In general, a better solution could be to choose this timing in a more dynamic fashion. For example, if the future path already has been aligned, then the switch from AC2F to S2C could trigger instantly when a distractor's health reaches zero. If alignment has not happened on the other hand, the switch could be delayed like it currently is to have some additional time for finishing the alignment. As far as the distance cooldown for distractors is concerned, it is likely best to start counting the cooldown as soon as the software detects that the player has started to move again. This cooldown could be triggered by using a movement threshold to check whether the player has started to move or not.

If AC2F as an algorithm was to support curvature gains in the future, then this would not require any changes to the timing of algorithm switches. At that point though, it might not be necessary to use S2C as an algorithm as AC2F could take care of everything. If AC2F supported curvature gains,

then it likely would have been closer to the mentioned "modified S2C" that Peck et al. as well as Chen and Fuchs have mentioned in their work [11, 30, 31]. This addition would of course increase the complexity of the algorithm, which is why this was not implemented in this thesis as it was outside of the allocated resource budget.

7.4 Effectiveness of Integration From a Game Design Point of View

While Ensemble Retriever's fully integrated distractors have been reasonably effective in terms of reorienting players away from physical walls, it is also worth to look at the integration from a game design point of view.

In general, participants did seem to enjoy the experience. Some excerpts from when participants were asked whether they enjoyed their experience includes:

- "Yes, it was very fun to play! I enjoyed the gameplay as it was quite original."
- "Yes, very enjoyable."
- "Quite fun, would be interesting to see multiple enemies at a time."

General feedback mentioned that the enemies in Experiment 1 were a bit slow. This issue was addressed in Experiment 2 by increasing the speed of attacks for relevant distractors and was positively received by the participants that took part in both experiments.

Despite this feedback, it was noticeable that participants got somewhat tired of facing the same distractors again and again towards the end of the experience. The frequency of distractors relative to the length of time spent on fighting them could also be a factor in this. This problem could be mitigated by having a more extensive variety of distractors that players can fight instead of just five varieties. Furthermore, introducing other types of distractors which are different from the battle ones could also decrease the monotony towards the end.

A more immediate solution that would not cost as many resources though would be to decrease the time spent on fighting distractors as mentioned in Section 6.3.4. While the distractor battles likely are the most engaging part of the experience, the time taken to reach the portal is relatively long. This time duration is of course a limitation with having to perform experiment measurements, and in a real-world scenario the current implementation could have worked better if the overall walking distance was shorter.

From a game design point of view, a shorter experience would likely work better and in particular if the player does not have both their shield and baton at the maximum level by the end. This approach would emphasise player choice as players would need to choose their upgrades more carefully since they would not have all of them when fighting the Mountain King. By doing so, it would allow players to either specialise in offence or defence depending on their personal preferences. While the current implementation allows the player to choose between upgrades when levelling up, the choices do not matter much as they will have a high enough level to unlock all the upgrades later on. Another option to further emphasise player choice would be to allow one final specialisation upgrade for either the shield or baton. The specialisation could in this case only be allowed to choose once regardless of how high the player's level is. This way, the player would need

to choose which of their tools receives the final upgrade.

7.5 Practical Challenges for Distractors in Redirected Walking

There has been a variety of practical challenges with distractors that have been observed while conducting Experiment 1 and 2. This section discusses these and provides some potential solutions.

7.5.1 Movement During Battles

While most participants stood relatively still while fighting distractors, some also moved around quite a bit. This movement creates some challenges as distractor battles always will be roughly 1m away from any reset bounds and 1.5m away from physical walls. If a participant moves around a lot, they might end up hitting a reset boundary which would be somewhat distracting to deal with in the heat of battle. While distractors and projectiles are paused during resets, it could still be considered as unwanted behaviour. The question then becomes: how do we limit the movement of certain players while they battle distractors? One potential option could be to make better use of deterrents to avoid having any players move further than 1m away from where their battle started. For example, certain objects could be strategically placed like the fire walls in Chen and Fuchs' research [30, 31] to deter the player from moving further. This approach was ultimately out of scope for the development of Ensemble Retriever, but could be considered as a potentially useful solution.

7.5.2 Stopping Speeds

Another practical challenge is the buffer between reset and distractor trigger boundaries with respect towards walking and stopping speeds. Ideally, the player should have as much walkable space as possible without triggering a distractor and then an immediate reset. While a 1m buffer between the two bounds has worked relatively well in Experiment 2, there were still some cases where the buffer was too small. In a similar fashion to dealing with movement during battles, it might be possible to further increase the stopping speed of participants with effective use of deterrents. As an example: the contrabass distractor spawns and immediately boxes the player into some sort of battle arena. The player might then end up stopping a bit faster compared to the current situation where the distractor simply spawns a few metres in front of them. While it is not guaranteed that this would solve the problem, it should at least be expected that players stop slightly faster to avoid crashing into a close virtual obstacle.

7.5.3 Concrete Distractors in Confined Spaces

Finally, one apparent problem with concrete distractors that make use of much movement is their use in more narrow or confined spaces. In Ensemble Retriever, this becomes an issue when fighting a distractor while close to the portal. In this case, there are potential situations where the distractor might end up moving into or behind the scenery as it rotates around the player. This behaviour is of course not wanted as it obscures vision towards the enemy and makes it harder to play. The question then becomes: how do we deal with distractors when the space around the player is more confined or narrow?

This could potentially be solved by analysing and letting the distractors know when they are in a more narrow area. With this additional knowledge, the distractor could for example move slightly closer to the player or limit their movement abilities to better work with their current environment. Finding a generic solution in this case is probably hard given the context-sensitive nature of fully integrated distractors. While generic solutions may be challenging to create for this, there are context-specific optimisations and solutions that can be developed. Outside of slightly more dynamic movement behaviour, it could be possible to have a set of distractors which specifically are aimed to work in more narrow spaces. If the game then detects that the player needs a distractor and is in a narrow space, it can spawn from this alternative distractor list to better suit the specific area. This way, it could also be possible to optimise how much the player moves their head around despite the limitations of the narrower space around them.

7.6 Participant Feedback on Pause - Turn - Centre

While it was not directly measured or focused on, some participants gave oral feedback on the Pause - Turn - Centre resetter. The general feedback was positive in that it felt natural and easy to use whenever needed, although it took some time to get used to the resetter initially. Participants who moved a lot during battles managed to trigger a few resets while fighting. They mentioned that it was rather disruptive and disorienting when it happened, which is to be expected. As far as the various goals of Pause - Turn - Centre are concerned, it managed to safely reorient participants away from physical walls and did not result in any issues in terms of HMD cable wrapping. As far as participant feedback is concerned, none mentioned the reset as causing cybersickness. This information is of course anecdotal though and would require a separate experiment to properly measure. In general though, no negative feedback was given in relation to Pause - Turn - Centre.

8 Conclusion and Future Work

This chapter includes the conclusion of the thesis and ends with the many different pathways that future work can take within the topic of distractors and redirected walking as a whole.

8.1 Conclusion

To summarise, this thesis has focused on exploring various aspects of noticeability and effectiveness when using fully integrated distractors. The methods that were used for this were quite exploratory as the space of distractor research still is relatively unexplored. As a result, a fair amount of future work has been identified. As part of the exploration, the Ensemble Retriever VR game has been developed which makes use of fully integrated distractors. The source code for Ensemble Retriever is openly available [42], making it valuable for both researchers and developers alike who wish to work with redirected walking. Together with Ensemble Retriever, the "Align Centre to Future" (AC2F) redirection algorithm has been developed as an alternative to "Steer to Center" (S2C) when standing relatively still. Furthermore, a new resetter: "Pause - Turn - Centre" has been developed to deal with the various shortcomings of existing resetting techniques.

The Ensemble Retriever game has been used to conduct two experiments. The first of these focused on seeing whether there were any significant differences in redirection noticeability between two states: A general walking state and a battle state against an enemy distractor. The experiment did not find any significant differences in detection thresholds for positive rotation gains. On the other hand, it was significantly easier for participants to notice negative rotation gains when fighting enemy distractors. This challenges prior expectations that distractors could make it more difficult to notice redirection [2, 18, 19] and warrants further research. Furthermore, an adaptation effect towards positive rotation gains has been observed which falls in line with Bölling et al.'s speculation that adaptation effects are possible for rotation gains [21]. This effect was not observed for negative rotation gains though.

The second experiment focused on testing the effectiveness of the AC2F algorithm. In this case, a condition where S2C was employed when walking and AC2F was employed during distractor battles was compared to a control condition where S2C was used for both scenarios. The results showed that there was no significant difference in the mean number of resets that occurred between the two conditions. No significant difference could be found in terms of mean time needed for a successful alignment between the participants' future virtual path and the physical room centre either. Despite this, the experimental condition with S2C+AC2F resulted in 15.8% fewer failure cases in terms of successful alignments towards physical space centre. This discrepancy has provided some valuable speculation for situations where the S2C algorithm could perform better than expected as well as potential optimisations for the AC2F algorithm during straight path walking scenarios.

8.2 Future Work

While this section focuses on areas of future research that were detected through discussion and results, it may not guarantee that there are not already any existing papers or research looking into this. The literature sample for this thesis is not comprehensive enough to cover the entirety of the redirected walking space, but serves as an approximation to the main topic of distractors within this space. The exploratory approach of this thesis means that many potential variables were at play due to the more realistic implementation and integration of distractors. As such, smaller and more controlled experiments on individual variables would be beneficial to fully understand the effects of some variables on others.

8.2.1 Experiment/Software Specific Future Work

This section contains potential future work that is directly related to the experiments and developed software in this thesis.

Improvements to AC2F

The first potential future work would be to further extend and improve the AC2F algorithm. In the current state, it is primarily used when the user is standing still and interacting with a distractor. By extending it with the addition of curvature gains and support for movement, the algorithm will be closer to what prior research has mentioned for their solutions [11, 30, 31]. The lack of available source code or full implementation details makes it challenging to approximate or implement the same solution as existing work has done. As such, any extensions to AC2F should remain openly available for the sake of future research and development.

There is also work that can be done to improve the smoothing functionality of AC2F. A current limitation with the algorithm is the somewhat sliding rotation effect which can happen during head rotations that do not use the body. Being able to identify these cases could allow for some dynamic changes to smoothing behaviour. Another option would be to re-evaluate how smoothing is applied and finding better means to do so.

Experiment Improvements

When looking back at the experiments themselves, there are improvements that could be made on the data recording end. A large amount of data post-processing was needed to acquire the necessary data to test all the hypotheses in Experiment 1 and 2. This data post-processing could technically be automated and handled on the software side. Furthermore, some improvements can be made to the accuracy of recording certain variables. These are further discussed in Section 6.3.3.

Evaluation of "Pause - Turn - Centre" in Respect to Other Resetters

Finally, there is additional evaluation that can be performed on the Pause - Turn - Centre resetter, which was introduced in this thesis. There are a variety of factors which could see further comparisons with other existing reset techniques. These include:

- The cybersickness effects of Pause - Turn - Centre vs. existing resetters.
- The safety of using Pause - Turn - Centre vs. existing resetters.

- The intrusiveness of using Pause - Turn - Centre vs. existing resetters.
- The effective distance travelled between resets when using Pause - Turn - Centre vs. existing resetters. This is something which could be measured with simulations in the Redirected Walking Toolkit.
- How much users become entangled in their HMD tether cables when using Pause - Turn - Centre vs. existing resetters.

8.2.2 Future Work for General Redirected Walking Research

This section contains potential future work which is relevant to redirected walking research as a whole.

The Asymmetry Between Positive and Negative Rotation Gains

The asymmetry between positive and negative rotation gains is a relatively interesting phenomenon which warrants further investigation. Additional background from psychology or neuroscience might help with enlightening what cognitive processes result in this difference. There could for example be some goal-oriented processes that result in the asymmetry. In this case, positive rotation gains will meet the goal looking direction and exceed it while negative gains fail to meet the goal. This expectation mismatch could have some effect, but further research is necessary to understand why and whether this is the case.

Further Research on Variables That Could Affect Noticeability

Given the results from Experiment 1, there may be additional unaccounted for variables that affected the measurements. Further research to identify new variables that affect noticeability would be beneficial. Having a frame of reference and its relation to optical flow is for example one area which could be investigated more deeply. Can a moving distractor in this case be considered as a frame of reference and does it make it easier to notice redirection? The results in this thesis seem to suggest so for negative rotation gains, but it is hard to draw any real conclusions due to the large data variability. Some participants have also mentioned using the glowing mushrooms as a frame of reference to more easily detect redirection. This information is of course anecdotal, but may be a similar factor to the presence of a salient distractor. Another variable could be differences in smoothing solutions between algorithms. In cases where a redirected walking solution uses multiple redirection algorithms, there may be minute differences in their smoothing solutions which may have some effect on noticeability.

How Smoothing Solutions Affect Redirection Noticeability

Given that there are various methods to implement smoothing for redirection algorithms, it would be worth investigating how different smoothing solutions affect the noticeability of redirection. By analysing and comparing different methods, it may be possible to find best practices in regards to what type of smoothing to implement. In this case, it is crucial to consider what methods provide the best user comfort and any potential noticeability effects.

Optimising Alignment Heuristics for Straight Walking Path Scenarios

In scenarios where a generic redirection algorithm like S2C is used in conjunction with a distractor specific counterpart, there are scenarios where the heuristic for the distractor algorithm can be optimised. The results in this thesis showed that employing AC2F together with S2C instead of a pure S2C solution resulted in more successful alignments towards the centre of the physical space. Despite this result, the pure S2C solution managed to perform similarly in terms of the two primary effectiveness metrics. This may be a result of it not always being ideal to align towards the physical room centre in straight walking path cases. Depending on where in the room the user is standing, an alternative heuristic might be more efficient. The pure S2C solution could in this case have achieved this more frequently by failing to fully align towards the physical centre. This case is further discussed in Section 6.3.2. Optimising this alignment heuristic so that it does not always rely on the centre of the room would be an interesting pathway for future research. This optimisation would be a good potential use case for the simulation functionality in the Redirected Walking Toolkit, and could be used to simulate whether this optimisation improves effectiveness or not.

Effects of Temporarily Disabling Gains

While an adaptation effect was experienced for positive rotation gains in Experiment 1, it should be noted that gains were disabled at certain times throughout the experience. Once a distractor has aligned the user's future path towards the room centre, gains are disabled to keep the wanted orientation stable. Given that participants on average spent ~20 seconds without any gains enabled during distractor battles, it may have had some effect on the strength of adaptation. This situation could for example be the reason why adaptation only was observed for positive rotation gains and not negative. In general, further research is necessary to understand how temporarily disabling redirection gains affect noticeability and potential adaptation effects.

Shorter Time Bursts of High Gains vs. Lower Gains For a Longer Duration

A related topic in terms of temporary gain disabling is the potential differences in comfort and cybersickness. It would be interesting to see further research on how a short burst of high gains with a period of disabled gains fares against a lower, but more prolonged exposure to gains. It is likely that these will affect cybersickness in some manner and finding whether there is a difference and if so, how large it is could benefit the future of redirected walking. This way, we can further inform our decisions on how we employ redirection in developed experiences.

Effectiveness of Disabling Redirection Towards the End of a Virtual Experience

Another element which could benefit from further research is the concept of disabling redirection gains towards the end of a virtual experience. The primary reasoning behind this approach is to allow the user to get accustomed to normal head rotations before taking off their HMD. How effective or whether this approach is effective at all for mitigating disorientation effects requires further investigation.

Whether Prior VR Experience Affects Noticeability

The results in this thesis suggest that prior VR experience affects how noticeable redirection is. Future research could test this with additional empirical data. It introduces some interesting questions like if a user's first few experiences with VR is with redirected walking only, will they adapt and normalise this as normal VR behaviour? If so, can they be further redirected than a regular user over time and how will they be affected by being exposed to normal head rotation in VR later on?

Effective and Short Estimation of Individual Detection Thresholds

There were large individual variances between participants in terms of detection thresholds which were seen in the results of Experiment 1. As such, there is room to improve and optimise how we estimate detection thresholds as it would be necessary to tailor gains to each individual's detection threshold. Conventional methods that take 30+ minutes would not be feasible in the real world as users should not be expected to spend this much time on calibration alone. The primary optimisation for estimation that is needed in this case is time.

A cursory glance through the literature databases shows that at least there is some research on the topic of individualised calibration [56]. Despite this, the research itself does not appear to be openly available and as such, makes it challenging to see how far this sub-field is developed.

8.2.3 The Future of Redirected Walking

Redirected walking as a field of research is continuously evolving. In the current day, consumer grade hardware for VR is at a level of quality where redirected walking can see usage on a larger scale than before. Despite this usage potential, much of the work in this field is still in the theoretical and academical space where projects are of small and highly controlled scopes. To further provide more nuance into this field, it is necessary to see more perspectives from the development and user ends by integrating redirected walking into larger projects. By doing so, more realistic and practical scenarios unfold which can contribute to providing new questions and insights which have not been seen before. Furthermore, it can result in new and innovative experiences for the end user. This thesis is just one small stepping stone in this direction. As both this work and existing literature shows, distractors are a promising means to the end of creating more immersive and engaging virtual experiences that seamlessly integrate redirected walking. With this integration, it becomes a natural extension of how interaction with virtual reality allows for an elevated subjective sense of presence [4, 5] and provides an effective means of locomotion [1, 5].

Bibliography

- [1] Razzaque, S., Kohn, Z., & Whitton, M. C. 2001. Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, 105–106. Citeseer.
- [2] Steinicke, F., Bruder, G., Jerald, J., Frenz, H., & Lappe, M. Jan 2010. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1), 17–27. doi:10.1109/TVCG.2009.62.
- [3] Azmandian, M., Grechkin, T., Bolas, M., & Suma, E. 2015. Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance. In *ICAT-EGVE 2015 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, Imura, M., Figueroa, P., & Mohler, B., eds. The Eurographics Association. doi:10.2312/egve.20151315.
- [4] Peck, T. C., Fuchs, H., & Whitton, M. C. May 2009. Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 15(3), 383–394. doi:10.1109/TVCG.2008.191.
- [5] Peck, T. C., Fuchs, H., & Whitton, M. C. March 2011. An evaluation of navigational ability comparing redirected free exploration with distractors to walking-in-place and joystick locomotion interfaces. In *2011 IEEE Virtual Reality Conference*, 55–62. doi:10.1109/VR.2011.5759437.
- [6] Fuglestad, B. N. Redirected walking, an investigation into noticeable, but usable gains and the role of hardware in threshold detection. Master's thesis, NTNU, 2018.
- [7] Suma, E. A., Bruder, G., Steinicke, F., Krum, D. M., & Bolas, M. March 2012. A taxonomy for deploying redirection techniques in immersive virtual environments. In *2012 IEEE Virtual Reality Workshops (VRW)*, 43–46. doi:10.1109/VR.2012.6180877.
- [8] Langbehn, E., Lubos, P., Bruder, G., & Steinicke, F. April 2017. Bending the curve: Sensitivity to bending of curved paths and application in room-scale vr. *IEEE Transactions on Visualization and Computer Graphics*, 23(4), 1389–1398. doi:10.1109/TVCG.2017.2657220.
- [9] Razzaque, S. 2005. *Redirected walking*. University of North Carolina at Chapel Hill.
- [10] Hodgson, E. & Bachmann, E. April 2013. Comparing four approaches to generalized redirected walking: Simulation and live user data. *IEEE Transactions on Visualization and Computer Graphics*, 19(4), 634–643. doi:10.1109/TVCG.2013.28.

- [11] Peck, T. C., Fuchs, H., & Whitton, M. C. March 2010. Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In *2010 IEEE Virtual Reality Conference (VR)*, 35–38. doi:10.1109/VR.2010.5444816.
- [12] Nguyen, A., Rothacher, Y., Kunz, A., Brugger, P., & Lenggenhager, B. March 2018. Effect of environment size on curvature redirected walking thresholds. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 645–646. doi:10.1109/VR.2018.8446225.
- [13] Steinicke, F., Bruder, G., Ropinski, T., & Hinrichs, K. 2008. Moving towards generally applicable redirected walking. In *Proceedings of the Virtual Reality International Conference (VRIC)*, 15–24. IEEE Press.
- [14] Kruse, L., Langbehn, E., & Steinicke, F. March 2018. I can see on my feet while walking: Sensitivity to translation gains with visible feet. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 305–312. doi:10.1109/VR.2018.8446216.
- [15] Waldow, K., Fuhrmann, A., & Grünvogel, S. M. March 2018. Do textures and global illumination influence the perception of redirected walking based on translational gain? In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 717–718. doi:10.1109/VR.2018.8446587.
- [16] Norouzi, N., Bruder, G., & Welch, G. 2018. Assessing vignetting as a means to reduce vr sickness during amplified head rotations. In *Proceedings of the 15th ACM Symposium on Applied Perception, SAP '18*, 19:1–19:8, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3225153.3225162>, doi:10.1145/3225153.3225162.
- [17] Neth, C. T., Souman, J. L., Engel, D., Kloos, U., Bühlhoff, H. H., & Mohler, B. J. March 2011. Velocity-dependent dynamic curvature gain for redirected walking. In *2011 IEEE Virtual Reality Conference*, 151–158. doi:10.1109/VR.2011.5759454.
- [18] Schmitz, P., Hildebrandt, J., Valdez, A. C., Kobbelt, L., & Ziefle, M. April 2018. You spin my head right round: Threshold of limited immersion for rotation gains in redirected walking. *IEEE Transactions on Visualization and Computer Graphics*, 24(4), 1623–1632. doi:10.1109/TVCG.2018.2793671.
- [19] Sra, M., Xu, X., Mottelson, A., & Maes, P. 2018. Vmotion: Designing a seamless walking experience in vr. In *Proceedings of the 2018 Designing Interactive Systems Conference, DIS '18*, 59–70, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3196709.3196792>, doi:10.1145/3196709.3196792.
- [20] Nguyen, A., Rothacher, Y., Lenggenhager, B., Brugger, P., & Kunz, A. 2018. Individual differences and impact of gender on curvature redirection thresholds. In *Proceedings of the 15th ACM Symposium on Applied Perception, SAP '18*, 5:1–5:4, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3225153.3225155>, doi:10.1145/3225153.3225155.

- [21] Bölling, L., Stein, N., Steinicke, F., & Lappe, M. May 2019. Shrinking circles: Adaptation to increased curvature gain in redirected walking. *IEEE Transactions on Visualization and Computer Graphics*, 25(5), 2032–2039. doi:10.1109/TVCG.2019.2899228.
- [22] Grechkin, T., Thomas, J., Azmandian, M., Bolas, M., & Suma, E. 2016. Revisiting detection thresholds for redirected walking: Combining translation and curvature gains. In *Proceedings of the ACM Symposium on Applied Perception, SAP '16*, 113–120, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/2931002.2931018>, doi:10.1145/2931002.2931018.
- [23] Mousavi, M., Jen, Y. H., & Musa, S. N. B. 12 2013. A review on cybersickness and usability in virtual environments. In *Current Trends in Ergonomics*, volume 10 of *Advanced Engineering Forum*, 34–39. Trans Tech Publications Ltd. doi:10.4028/www.scientific.net/AEF.10.34.
- [24] Rietzler, M., Gugenheimer, J., Hirzle, T., Deubzer, M., Langbehn, E., & Rukzio, E. Oct 2018. Rethinking redirected walking: On the use of curvature gains beyond perceptual limitations and revisiting bending gains. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 115–122. doi:10.1109/ISMAR.2018.00041.
- [25] Hildebrandt, J., Schmitz, P., Calero Valdez, A., Kobbelt, L., & Ziefle, M. 2018. Get well soon! human factors' influence on cybersickness after redirected walking exposure in virtual reality. In *Virtual, Augmented and Mixed Reality: Interaction, Navigation, Visualization, Embodiment, and Simulation*, Chen, J. Y. & Fragomeni, G., eds, 82–101, Cham. Springer International Publishing.
- [26] Suma, E. A., Clark, S., Krum, D., Finkelstein, S., Bolas, M., & Warte, Z. March 2011. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*, 159–166. doi:10.1109/VR.2011.5759455.
- [27] Williams, B., Narasimham, G., Rump, B., McNamara, T. P., Carr, T. H., Rieser, J., & Bodenheimer, B. 2007. Exploring large virtual environments with an hmd when physical space is limited. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization, APGV '07*, 41–48, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/1272582.1272590>, doi:10.1145/1272582.1272590.
- [28] Strauss, A. & Corbin, J. 1994. Grounded theory methodology. *Handbook of qualitative research*, 17, 273–85.
- [29] Nielsen, L. T., Møller, M. B., Hartmeyer, S. D., Ljung, T. C. M., Nilsson, N. C., Nordahl, R., & Serafin, S. 2016. Missing the point: An exploration of how to guide users' attention during cinematic virtual reality. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology, VRST '16*, 229–232, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/2993369.2993405>, doi:10.1145/2993369.2993405.

- [30] Chen, H. & Fuchs, H. 2017. Towards imperceptible redirected walking: Integrating a distractor into the immersive experience. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '17, 22:1–22:2, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3023368.3036844>, doi:10.1145/3023368.3036844.
- [31] Chen, H. & Fuchs, H. 2017. Supporting free walking in a large virtual environment: Imperceptible redirected walking with an immersive distractor. In *Proceedings of the Computer Graphics International Conference*, CGI '17, 22:1–22:6, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3095140.3095162>, doi:10.1145/3095140.3095162.
- [32] Rothe, S. & Hußmann, H. 2018. Guiding the viewer in cinematic virtual reality by diegetic cues. In *Augmented Reality, Virtual Reality, and Computer Graphics*, De Paolis, L. T. & Bourdot, P., eds, 101–117, Cham. Springer International Publishing.
- [33] Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., Masia, B., & Wetzstein, G. April 2018. Saliency in vr: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4), 1633–1642. doi:10.1109/TVCG.2018.2793599.
- [34] Grechkin, T., Azmandian, M., Bolas, M., & Suma, E. March 2015. Towards context-sensitive reorientation for real walking in virtual reality. In *2015 IEEE Virtual Reality (VR)*, 185–186. doi:10.1109/VR.2015.7223357.
- [35] Nilsson, N. C., Peck, T., Bruder, G., Hodgson, E., Serafin, S., Whitton, M., Steinicke, F., & Rosenberg, E. S. Mar 2018. 15 years of research on redirected walking in immersive virtual environments. *IEEE Computer Graphics and Applications*, 38(2), 44–56. doi:10.1109/MCG.2018.111125628.
- [36] Sun, Q., Patney, A., Wei, L.-Y., Shapira, O., Lu, J., Asente, P., Zhu, S., Mcguire, M., Luebke, D., & Kaufman, A. July 2018. Towards virtual reality infinite walking: Dynamic saccadic redirection. *ACM Trans. Graph.*, 37(4), 67:1–67:13. URL: <http://doi.acm.org/10.1145/3197517.3201294>, doi:10.1145/3197517.3201294.
- [37] Langbehn, E., Steinicke, F., Lappe, M., Welch, G. F., & Bruder, G. July 2018. In the blink of an eye: Leveraging blink-induced suppression for imperceptible position and orientation redirection in virtual reality. *ACM Trans. Graph.*, 37(4), 66:1–66:11. URL: <http://doi.acm.org/10.1145/3197517.3201335>, doi:10.1145/3197517.3201335.
- [38] Azmandian, M., Grechkin, T., Bolas, M., & Suma, E. March 2016. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *2016 IEEE 2nd Workshop on Everyday Virtual Reality (WEVR)*, 9–14. doi:10.1109/WEVR.2016.7859537.
- [39] Nguyen, A. & Kunz, A. 2018. Discrete scene rotation during blinks and its effect on redirected walking algorithms. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software*

- and Technology*, VRST '18, 29:1–29:10, New York, NY, USA. ACM. URL: <http://doi.acm.org/10.1145/3281505.3281515>, doi:10.1145/3281505.3281515.
- [40] Malloy, K. M. & Milling, L. S. 2010. The effectiveness of virtual reality distraction for pain reduction: A systematic review. *Clinical Psychology Review*, 30(8), 1011 – 1018. URL: <http://www.sciencedirect.com/science/article/pii/S0272735810001091>, doi:<https://doi.org/10.1016/j.cpr.2010.07.001>.
- [41] Knight, K., McClenaghan, C. E., & Singh, B. 2019. Virtual reality distraction from painful procedures in the paediatric emergency department. *Archives of Disease in Childhood*, 104(2), 204–205. URL: <https://adc.bmj.com/content/104/2/204.2>, arXiv:<https://adc.bmj.com/content/104/2/204.2.full.pdf>, doi:10.1136/archdischild-2018-315921.
- [42] Ensemble Retriever - Project Repository. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository>. (Last Visited on 27.05.2019).
- [43] Takahashi, A. Mersenne Twister Library For C#. <https://www.nuget.org/packages/MersenneTwister/>. (Last Visited on 27.05.2019).
- [44] Project Repository - RedirectionManagerER Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/RedirectedWalking/RedirectionManagerER.cs>. (Last Visited on 27.05.2019).
- [45] Project Repository - DistractorEnemy Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/Distractors/DistractorEnemy.cs>. (Last Visited on 27.05.2019).
- [46] Project Repository - AC2FRedirector Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/RedirectedWalking/Redirectors/AC2FRedirector.cs>. (Last Visited on 27.05.2019).
- [47] Unity Forums - Thread with interpolation solution for AC2F. <https://forum.unity.com/threads/how-to-smooth-damp-towards-a-moving-target-without-causing-jitter-in-the-movement.130920/>. (Last Visited on 09.05.2019).
- [48] Project Repository - Pause-Turn-Centre Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/RedirectedWalking/Resetters/PauseTurnCentre.cs>. (Last Visited on 27.05.2019).
- [49] Project Repository - ExperimentDataManager Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/ExperimentRelated/ExperimentDataManager.cs>. (Last Visited on 27.05.2019).
- [50] Project Repository - GainIncrementer Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/Scripts/ExperimentRelated/GainIncrementer.cs>. (Last Visited on 27.05.2019).

- [51] Project Repository - Updated Version of HeadFollower Script. <https://github.com/hedvik/EnsembleRetriever-ProjectRepository/blob/master/Assets/RDW%20Toolkit/Scripts/Misc/HeadFollower.cs>. (Last Visited on 27.05.2019).
- [52] Wang, A. YouTube - Full Playthrough of Ensemble Retriever. <https://youtu.be/LjyxvXxf01c>. (Last Visited on 27.05.2019).
- [53] Saito, M. & Matsumoto, M. 2008. Simd-oriented fast mersenne twister: a 128-bit pseudo-random number generator. In *Monte Carlo and Quasi-Monte Carlo Methods 2006*, Keller, A., Heinrich, S., & Niederreiter, H., eds, 607–622, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [54] Shapiro, S. S. & Wilk, M. B. 1965. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4), 591–611. URL: <http://www.jstor.org/stable/2333709>.
- [55] Mann, H. B. & Whitney, D. R. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1), 50–60. URL: <http://www.jstor.org/stable/2236101>.
- [56] Hutton, C., Ziccardi, S., Medina, J., & Rosenberg, E. S. 2018. Individualized Calibration of Rotation Gain Thresholds for Redirected Walking. In *ICAT-EGVE 2018 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, Bruder, G., Yoshimoto, S., & Cobb, S., eds. The Eurographics Association. doi:10.2312/egve.20181315.
- [57] Azmandian, M. YouTube - Redirected Walking Toolkit Setup. <https://www.youtube.com/watch?v=9agwHrNRfM>. (Last Visited on 16.05.2019).

A Terminology and Abbreviations

VR - Virtual Reality.

VE - Virtual Environment.

RDW - Redirected Walking.

ROT - Reorientation Technique. These are used in redirected walking whenever the user is close to leaving the physical tracking space in order to reorient them away from it.

Reset Technique - A type of ROT. These forcefully reorient the user away from physical walls. While they can break the subjective sense of presence, they are primarily used as fail-safes if other methods are insufficient.

NPC - Non-playable character. A term used in video games to classify a computer-controlled entity.

HUD - Heads-Up Display.

GUI - Graphical User Interface.

S2C - "Steer To Center". A redirection algorithm that applies gains in a manner that redirects the user towards the centre of the tracking space.

S2O - "Steer To Orbit". A redirection algorithm that applies gains in a manner that redirects the user to walk on the edge of a circle in the tracking space.

AC2F - "Align Centre To Future". A redirection algorithm that has been developed for this thesis. It focuses on redirecting a standing user so their future walking direction is aligned with the centre of the physical space.

FoV - Field of View.

HMD - Head Mounted Display. Also often known as a virtual reality headset.

IDE - Integrated Development Environment.

2AFC - Two-alternative Forced Choice.

B Reproducing The Experiments

This appendix provides the necessary steps to reproduce the environment and experiments which were used in this thesis. The appendix is aimed to be fairly self-contained, so it may duplicate some already mentioned information from previous chapters. It is meant to function as a serial guide to how the experiments themselves can be reproduced.

B.1 Before Starting

Before providing information on how to reproduce the experiments, it is necessary to understand the environments that they were conducted in. This section summarises how the physical room was set up in conjunction with software and hardware.

B.1.1 Room Setup

This thesis made use of a physical room with a 5m x 5.75m size. While reproducing the exact dimensions is not necessary, there are additional steps needed in case of other room dimensions. In general, a relatively square form is preferred due to how algorithms like S2C functions [3].

B.1.2 Hardware Setup

On the hardware side, a standard HTC Vive HMD was employed with its respective controllers. In order for the HMD to be able to reach through the entire physical space, a 5m HDMI cable, 5m USB 2.0 cable extender and 5m power cable extender were used. It is important to note that USB and HDMI cables are close to their physical limits at 5m. Therefore, not all cables may work in this situation. As such, it is important to research and test whether the chosen cables and cable extenders are functioning with the HMD beforehand. In order to provide audio, a pair of Audio Technica ATH-MSR7BK headphones were connected to the HMD mini-jack port.

The desktop computer which was used for both experiments consisted of the following technical specifications:

CPU: Intel i7-6700k

GPU: Nvidia Geforce GTX 1080

RAM: 16 GB

Operating System: Windows 10 Pro

B.1.3 Software Setup

On the software side, it is necessary to have the following software installed:

- Unity Engine, version 2018.3.5f1.

- SteamVR, version 1.3.23. Using the newest version should not be an issue though.
- OpenVR Advanced Settings¹ is recommended to disable the HTC Vive’s internal chaperone system.
- The Ensemble Retriever Unity project which can be directly cloned from the project repository [42].

B.1.4 The Ensemble Retriever Project

When first opening the Ensemble Retriever project it is necessary to choose one of the two experiment scenes. These can be found under the *Assets/Scenes/* folder. Both scenes have the same scene hierarchy with some minor differences in parameters and positions of game objects. For the sake of providing examples, this section will assume that *Assets/scenes/Experiment2Scene* has been chosen.

The primary objects that will be discussed in the scene consist of the following:

- "ExperimentManager"
- "Redirected Walker (Debug)" and "Redirected Walker (VR)"
- "GameManager"

ExperimentManager

The "ExperimentManager" object contains the *ExperimentDataManager* and *GainIncrementer* scripts.

ExperimentDataManager is used to define which experiment that will be performed and the names of resulting data recording files. Furthermore, there are some experiment specific variables. The most important one is the length of the buffer window containing all applied gains. This is used for the calculation of various variables in Experiment 1 which are detailed in Section B.2.2.

GainIncrementer allows the researcher to define start values for gains, their maximum/minimum values and various variables related to the randomness of the incrementer. A time step base is set in conjunction with time step noise. A time step base of 5 with a noise value of 2.5 means that gains will be incremented every 2.5-7.5 seconds. Similar variables exist for rotation and curvature gain increments.

Redirected Walker (Debug) and Redirected Walker (VR)

There are two redirection root objects within each scene: "Redirected Walker (Debug)" and "Redirected Walker (VR)". At any given time, only one of these objects should be active. The debug version allows for keyboard controls with the following key-binds:

W, A, S, D: Movement.

Arrow Keys: Rotation/looking direction.

T: Text box advancement.

L: Allows to choose the baton upgrade when levelling up.

K: Allows to choose the shield upgrade when levelling up.

¹<https://github.com/OpenVR-Advanced-Settings/OpenVR-AdvancedSettings>

Spacebar: Shooting with the conductor baton when charged.

O: Enables the nearest objective pointer.

Outside of this, the roots contain the following parameters which can be modified.

Redirection Gain Parameters: The strength of various redirection gains. These will be automatically be controlled during Experiment 1, while these have to be set to the desired values for Experiment 2. It should be noted that translation gains are not used by any of the employed algorithms.

Tracking Space Fade Speed: The speed of the animation for fading the representation of the physical space out or in during resetting.

Always Display Tracking Floor: Debug setting that will let the floor from the physical space representation stay outside of resets.

Switch To AC2F on Distractor Spawn: Toggle to enable AC2F when a distractor is active. It is automatically set during Experiment 2 depending on what condition the participant is assigned to.

Alignment Threshold: The dot product threshold for the amount of error is allowed in terms of aligning the future path towards the physical centre during distractor battles. -1 would be a perfect alignment while increased values allow for more error.

Distractor Magnitude Cooldown: This variable defines how large of a distance in metres a user needs to travel before a new distractor can trigger. This value **needs** to be individually tested for different room sizes as it may be too long for smaller sizes. It should also be noted that counting towards this distance only starts once the death animation of a distractor has finished.

Debug Distractor: If this field has a distractor object assigned it will always spawn this distractor for the sake of debugging.

Debug Gain Application Type: When toggled, the objective pointer's material will correspond to the gain that is currently being applied. The materials to represent gains can be set right below this setting. This is a debug feature aimed at checking which gain is applied at any time.

The attached AC2F redirector and Pause - Turn - Resetter also has some parameters:

AC2F - Super Smoothing Enabled: With this toggle active, the AC2F will use the implemented smoothing solution. The algorithm will default back to Azmandian et al.'s smoothing solution from their S2C implementation otherwise.

AC2F - Super Smooth Speed: Parameter to tweak the speed in seconds of the implemented smoothing solution.

Pause - Turn - Centre - Safety Mode: A toggle that makes the resetter finalise once the user is looking towards the centre of the room and is back inside safe bounds. If disabled, the resetter will finalise as long as the user is looking towards the room centre. This is not recommended as participants might hit a physical wall if moving backwards during battles. In this case, they might be looking towards the centre and hit a physical wall as the reset will not activate since its condition for whether a reset is necessary or not is true.

GameManager

The "GameManager" object contains a large amount of game-related parameters and data which is not necessary to detail for this section. Despite this, it is worth noting the "Skip Tutorial" and "Debug Mode" variables. The first of these toggles is self-explanatory while the second opens a wide variety of functionality. With debug mode enabled the following functionality is available:

- Pressing the grip buttons on the baton controller automatically puts the baton into a charged and attack ready state. If using the keyboard/debug redirected walker this functionality is available with the press of the "P" key.
- Pressing the grip button on the shield controller automatically levels up the player. If using the keyboard/debug redirected walker this functionality is available with the press of the "X" key.

Individual Room Calibration

Calibration of the physical space can be done on the "Tracked Space" object which is a child of the redirection root object. Azmandian et al. have provided a guide on how to do this on YouTube [57]. The previously mentioned distractor magnitude cooldown variable may also need to be modified for room sizes which are smaller than the one used for this thesis.

A recommended way to handle the calibration of the room itself is to run the Ensemble Retriever game and place each controller at one edge of the physical space. This way, it is possible to modify and scale the physical space representation accordingly as the controllers will be mapped to where the physical boundaries are.

B.2 Experiment 1

In order to perform Experiment 1, there are a few things which should be in place first:

- "Experiment1Scene" should be the currently active scene.
- "ExperimentDataManager" should be set to perform a detection experiment.
- "Redirected Walker (VR)" is enabled and its debug counterpart is disabled.

The following sections will detail the exact information participants were given, the procedure of the experiment, the data recording format and the data post-processing steps.

B.2.1 Participant Information and Procedure

The experiment starts with participants needing to provide consent. For this thesis, consent was handled in a written form. Once the participant has provided consent, the following information is

given:

1. Participants are told that they will be playing through a VR game that makes use of redirected walking for approximately 20-30 minutes.
 - If participants have not heard of redirected walking before they are given a brief introduction to the topic. Participants are informed that redirected walking will try to make small changes to the way they move and look around in VR so that they can be steered away from physical walls.
2. Participants are informed that they should not walk too quickly as the reset boundaries only trigger 0.5m away from any physical boundaries. For the same reason, they are discouraged from moving around too much during battles.
3. Participants are informed that they should be mindful of the HMD tether cable as they move around. They may need to untangle themselves at times.
4. Finally, participants are informed that if they at any times start to feel nauseous or cybersick, they should inform of this and stop the experiment immediately.

The participant is then allowed to put on the Vive HMD and is given their controllers and headphones. They are also encouraged to put their thumb on the detection button so they can quickly press it as soon as they notice any redirection. In this case, it is the menu button on their shield controller. The controls will be taught to the participant through the ingame tutorial. Once the participant has started the tutorial, the lights in the physical room should be dimmed to limit any potential reference points which can be gleaned from under the HMD.

The playthrough of Ensemble Retriever is finished when the participants' score has been displayed. At this point, the game can be turned off as all related data has been recorded, and participants can take off their HMD. The experiment ends with having participants answer the demographics questionnaire.

B.2.2 Data Recording/Data Format

The performance data for this experiment is recorded in a serial fashion per participant. In this case, each frame of Ensemble Retriever is recorded with a variety of different variables in a comma separated file format. This means that each row in the data can be considered as one frame with many column variables. This results in relatively large files, and it becomes important to understand what each recorded variable represents. This section provides an overview of these variables.

The first set of recorded variables are as follows:

ParticipantID: The unique identifier for the current participant. This is in an integer format and increments per participant.

GainDetected: Whether a gain was detected this frame. In particular, this value is set to 1 on the frame that the participant has pressed the detection button on their controller.

DeltaPosMagnitude: The distance magnitude of movement in metres that the participant has

moved between this and the prior frame.

DeltaDir: The angle that the participant's head has rotated on the vertical axis between this and the prior frame.

DeltaTime: The time that has passed in seconds between this and the prior frame.

After these variables, there are a large variety of variables dedicated to whether specific elements of Ensemble Retriever are active or not on the given frame. If something is active, it will have a value of 1, and 0 otherwise. These binary columns stop after the `CurvatureGainApplied` variable.

The next set of variables is the current strength of redirection gains. These will increase and decrease throughout the experiment as they accumulate and participants detect them:

CurrentRotationGainAgainst: Current negative rotation gain. Read as a percentage.

CurrentRotationGainWith: Current positive rotation gain. Read as a percentage.

CurrentCurvatureRadius: Current curvature radius. Read in metres.

The next set of variables only contain data when the "GainDetected" variable has a value of 1. These make use of the aforementioned buffer window from Section B.1.4 to provide the ratio of what gains were present as the participant detected a gain. By default, this will be the ratio of gains that were applied over the last half second before the detection button was pressed. The ratios are in the numerical range of 0-1 and represent a percentage. These ratio variables are named as follows:

- `NoGainRatioDuringDetection`.
- `NegativeRotationGainRatioDuringDetection`.
- `PositiveRotationGainRatioDuringDetection`.
- `CurvatureRotationGainRatioDuringDetection`.

The remaining data columns/variables are as follows:

MostLikelyDetectedGain: The value of the gain that had the largest ratio at the time of detection. Can be 0 in a few cases which are further discussed in Section 5.1.6.

TimeSinceExperimentStart: The time in seconds since the participant finished the tutorial and started playing.

AlgorithmCategory: Categorical variable for the currently active redirection algorithm. (0 = S2C, 1 = AC2F).

AppliedGainCategory: Categorical variable for the currently applied redirection gain. (-1 = None, 0 = Negative rotation, 1 = Positive rotation, 2 = Curvature).

DistractorCategory: Categorical variable for what current distractor is active. (0 = None, 1 = Contrabass, 2 = Oboe, 3 = Harpsichord, 4 = Violin, 5 = Glockenspiel).

MostLikelyDetectedGainCategory: Categorical variable for the type of gain that most likely was detected. Uses the same values and labels as "AppliedGainCategory".

B.2.3 Data Post Processing

In order to acquire the relevant data needed to finish Experiment 1, a post-processing step on the recorded data is necessary. For this thesis, IBM SPSS Statistics 25 was used. This section will primarily focus on the thought process for how to acquire the relevant variables without going too deeply into software specific methods.

Section 5.1.6 mentions four potential cases for when the "MostLikelyDetectedGain" variable has a value of 0. The following paragraphs will focus on how to process cases 1 and 4 into useable detection data. It should be noted that this post-processing step only is applied to rotation gain detections. It is recommended that this post-processing is applied to a copy of the processed variables for the sake of keeping the history of the original data intact.

Extracting Additional Rotation Gain Detections for Case 1

Case 1 scenarios are where the "MostLikelyDetectedGain" variable has a value of 0 for one specific reason. In this case, the participant has pressed the button to detect a rotation gain, but by the time they pressed the button, the AC2F algorithm had already finished the alignment. As a result, both rotation gains are disabled and set to a value of 0.

The fact that "CurrentRotationGainAgainst" and "CurrentRotationGainWith" are set to 0 during AC2F alignment is somewhat problematic for post-processing. To deal with this, the first step of post-processing is to create a copy of these variables that simply repeats the value of the previous frame during alignment rather than being a value of 0. This can be handled in a variety of ways, but for this thesis, values of 0 were set as missing values and replaced by an interpolation. Since the gain values are the same before and after alignment, this interpolation will repeat the gain values. This simplifies the rest of the post-processing steps. These generated variables will be referred to as the interpolated counterparts of "CurrentRotationGainAgainst" and "CurrentRotationGainWith" forwards.

There are two different approaches which can be used to extract correct rotation gains during case 1 scenarios:

Alternative 1: Using the "MostLikelyDetectedGainCategory" Variable

For the first and simple alternative it is first necessary to do a conditional data selection:

$$\begin{aligned} \textit{Selection} : & \textit{GainDetected} = 1 \wedge \textit{MostLikelyDetectedGain} = 0 \wedge \\ & \textit{CurrentRotationGainAgainst} = 0 \wedge \textit{MostLikelyDetectedGainCategory} \neq -1 \end{aligned}$$

For this selection, either "CurrentRotationGainAgainst" or "CurrentRotationGainWith" can be used as both variables will be 0 at the same time. This selection identifies case 1 scenarios where "MostLikelyDetectedGainCategory" suggests that an actual gain was detected. By using this variable, it is now possible to set the value of the extracted rotation gain with help from the interpolated

variables that were created in the prior step. If $MostLikelyDetectedGainCategory = 0$, then we set the value to the interpolated counterpart of "CurrentRotationGainAgainst". If

$MostLikelyDetectedGainCategory = 1$, then we set the value to the interpolated counterpart of "CurrentRotationGainWith".

Alternative 2: Using Ratio Variables

The second approach to dealing with case 1 scenarios is a bit more detailed. It starts with the following selection:

$$\begin{aligned}
 Selection : GainDetected = 1 \wedge \\
 MostLikelyDetectedGain = 0 \wedge CurrentRotationGainAgainst = 0 \wedge \\
 (NegativeRotationGainRatioDuringDetection \neq 0 \vee \\
 PositiveRotationGainRatioDuringDetection \neq 0)
 \end{aligned}$$

With this selection, we can check which of the two rotation gain ratios are highest. The extracted rotation gain can then be conditionally set to that of the interpolated counterpart to "CurrentRotationGainAgainst" or "CurrentRotationGainWith" depending on which ratio was highest. If $NegativeRotationGainRatioDuringDetection > PositiveRotationGainRatioDuringDetection$ then we set the value of the extracted rotation gain to the interpolated counterpart to "CurrentRotationGainAgainst". Else, it is set to the interpolated counterpart of "CurrentRotationGainWith".

Extracting Additional Rotation Gain Detections for Case 4

Case 4 scenarios are where the "MostLikelyDetectedGain" variable has a value of 0 for one specific reason. In this case, the participant has detected a rotation gain, but they pressed the button relatively late while their head was not moving. This results in the 0.5 ratio buffer to conclude that the dominant ratio was that no gains were applied and sets the value of "MostLikelyDetectedGain" to 0.

This case can be processed by using the second highest ratio in hopes of finding a remaining trace of the correct rotation gain in the ratio buffer. The following selection is used for this processing step:

$$\begin{aligned}
 Selection : GainDetected = 1 \wedge MostLikelyDetectedGain = 0 \wedge \\
 CurrentRotationGainAgainst \neq 0 \wedge NoGainRatioDuringDetection \neq 1
 \end{aligned}$$

Similarly to alternative 2 when processing case 1, the extracted rotation gain can be set by looking at which of the two rotation gain ratios is the highest.

B.3 Experiment 2

In order to perform Experiment 2, there are a few things which should be in place first:

- "Experiment2Scene" should be the currently active scene.
- "ExperimentDataManager" should be set to perform an effectiveness experiment.

- "Redirected Walker (VR)" is enabled and its debug counterpart is disabled.

The following sections will detail the exact information participants were given, the procedure of the experiment, the data recording format and the data post-processing steps.

B.3.1 Participant Information and Procedure

The procedure and information given to participants is equivalent to that mentioned in Section B.2.1 with some minor changes. The only differences are that the approximate time needed to finish the experiment is 15-20 minutes and no detection specific information is given.

B.3.2 Data Recording/Data Format

The data collection and format in Experiment 2 is similar to Experiment 1 (detailed in Section B.2.2). Data is recorded per frame throughout the Ensemble Retriever playthrough and consists of the following variables:

ParticipantID: The unique ID of the participant.

GroupID: The ID of the condition that the participant was randomly assigned to. (0 = S2C Only/- Control Group, 1 = S2C+AC2F/Experiment Group).

TimeSinceExperimentStart: The time in seconds since the participant finished the tutorial and started playing. Accumulates over time.

TimeSpentWalking: The time in seconds that the participant has spent walking. Accumulates over time while no distractors are active.

NumberOfResets: The number of resets that have happened so far. It is incremented on the frame that a reset has been triggered.

NumberOfDistractors: The number of distractors that have been triggered so far. It works similarly to NumberOfResets.

IsResetActive: A binary value that either is 0 if no reset was active on a given frame or a series of 1's for as long as a reset has been active.

IsDistractorActive: Similar to IsResetActive, but for distractors instead.

CurrentlyActiveDistractorType: Categorical variable for what current distractor is active. (0 = None, 1 = Contrabass, 2 = Oboe, 3 = Harpsichord, 4 = Violin, 5 = Glockenspiel)

AlignmentComplete: Binary variable that is set to 1 whenever the user's future path has been aligned with the centre of the physical space. The value is reset back to 0 after the distractor that resulted in the alignment has been defeated.

AlignedThisFrame: Set to 1 on the frame that alignment to the centre of the room has happened. This makes it simple to find all the times where the user has been aligned and gather additional information.

TimeTakenUntilAlignment: The time in seconds needed for a distractor to align the user's future path to the physical room centre. This variable only consists of a proper value when `AlignedThisFrame` is 1 and is 0 otherwise.

DistractorDefeatedThisFrame: Similar to `AlignedThisFrame`, but is set to 1 on the frame that a distractor has been defeated instead.

TimeTakenToDefeatDistractor: Consists of the time in seconds needed to defeat the currently active distractor. This variable only has a value other than 0 when `DistractorDefeatedThisFrame` is 1.

CurrentPlayerShieldLevel: The current level of the player's shield.

CurrentPlayerBatonLevel: The current level of the player's baton.

DeltaPos: The distance in metres that the participant has moved between frames. To be more specific, it is the magnitude of `DeltaPos`.

DeltaDir: The angle that the participant's head has rotated on the vertical axis between frames.

DeltaTime: The time between frames in seconds.

B.3.3 Data Post Processing

The data post-processing in Experiment 2 is divided into two major steps, one for each of the hypotheses that were tested. The first step is related to identifying all legitimate resets while discarding unintentional ones. The second step is related to finding legitimate alignments and adding penalties for failed alignments. Outside of these major steps, smaller aggregations were made on various variables for the sake of presentation. These are not detailed here as general aggregation is simple enough to not warrant a step by step process.

There are a variety of situations where it might be preferable to only select one row of data per participant. This is possible by selecting each row where `"TimeSinceExperimentStart"` has a value of 0 as it should uniquely contain this value at the start of each participants' data. There could of course be situations where this variable is not 0, so it is worth to check before doing further processing with this approach. It was consistent in the case of this thesis, but various hardware configurations and so on could affect this.

Finding Legitimate Resets

The post-processing for finding legitimate resets is divided into several smaller steps:

Step 1: Creating a Timer During Distractor Spawns

The first step in identifying legitimate resets is to create a timer variable which counts from 0 each time a distractor spawns. This can be handled by initialising the timer variable to a value of 0 for all rows. Using a LAG function, it is possible to check the value of a previous row, allowing for pattern detection. For the sake of this example, the new variable is called `"TimeSinceDistractorSpawn"`.

Before starting the calculation, it is necessary to make a selection:

$$\textit{Selection} : \textit{IsDistractorActive} = 1$$

The new timer variable can then be calculated in the following fashion:

$$\textit{TimeSinceDistractorSpawn} = \textit{LAG}(\textit{TimeSinceDistractorSpawn}) + \textit{DeltaTime}$$

This will process each frame where a distractor is active and accumulate a timer which resets back to 0 once the distractor is finished. The selection can then be disabled.

Step 2: Finding Frames With Legitimate Resets

For this step, a new variable to store legitimate resets is created. For the sake of examples, it will be called "LegitimateResetHappenedThisFrame" and initialised with a value of 0. This variable can then be set to 1 given the following condition:

$$\textit{If} : (\textit{IsResetActive} = 1 \wedge \textit{LAG}(\textit{IsResetActive}) = 0)$$

This will set the value of the variable to 1 on the frame that a reset has activated. The next step is then to remove unintentional resets, which can be handled by taking the previously computed variable and conditionally setting it to 0:

$$\begin{aligned} \textit{If} : & (\textit{IsResetActive} = 1 \wedge \textit{LAG}(\textit{IsResetActive}) = 0 \wedge \\ & \textit{IsDistractorActive} = 1 \wedge \textit{TimeSinceDistractorSpawn} \leq 10) \end{aligned}$$

This will remove all resets that happen within the first 10 seconds of a distractor spawn. The reasoning for choosing a specific value of 10 is mentioned in Section 6.1.3.

Step 3: Aggregate Legitimate Reset Sums

The final step is to aggregate the number of legitimate resets that each participant experienced. Using the previously computed "LegitimateResetHappenedThisFrame" variable, it is possible to aggregate a sum of legitimate resets by breaking the aggregate on "ParticipantID".

Finding Legitimate Alignments and Including Alignment Time Penalties

The second major post-processing step relates to finding legitimate alignments and adding alignment time penalties. Legitimate alignments are in this case defined as all alignments that happen without the help of any resets.

Finding Legitimate Alignments

As with most post-processing steps, this step starts with the creation of a new variable. For the sake of examples, this will be called "LegitimateAlignmentHappenedThisFrame" and initialise it to a value of 0. This variable is set to 1 if:

$$\textit{If} : (\textit{AlignedThisFrame} = 1 \wedge \textit{LAG}(\textit{IsResetActive}) \neq 1)$$

This will yield a value of 1 for each frame where an alignment happened without the aid of a reset.

Identifying Failed Alignments and Including Penalties

In order to identify failed alignments, a new variable is created and initialised to 0: "AlignmentFailedThisFrame". We can set this variable to a value of 1 in the following case:

$$If : DistractorDefeatedThisFrame = 1 \wedge LAG(AlignmentComplete) = 0$$

It is then possible to select all rows where "AlignmentFailedThisFrame" has a value of 1 and add a penalty time to "TimeTakenUntilAlignment". In the case of this thesis, a penalty time of 54.86 seconds was used as mentioned in Section 6.1.3. Including failure cases into the selection for the data sample then becomes as follows:

$$Selection : LegitimateAlignmentHappenedThisFrame = 1 \vee AlignmentFailedThisFrame = 1$$

With this selection, it is then possible to create aggregates or graphs out of the "TimeTakenUntilAlignment" variable. Since the selection includes failed alignments, it will create a respective skew if one condition has a larger amount of failed alignments compared to another.

C Demographics Questionnaire

This appendix includes the demographic questionnaire which was used for Experiment 1 and 2 in this thesis.

Demographical Questions (Required)

1. What is your participant ID?

2. What is your gender?

Mark only one oval.

- Female
- Male
- Prefer not to say
- Other: _____

3. What is your age range?

Mark only one oval.

- 18 - 24
- 25 - 34
- 35 - 44
- 45 - 54
- 55 - 64

4. If you have any optical corrections, did you need to remove them when using the VR headset?

Mark only one oval.

- Yes
- No

5. Have you taken part in any prior virtual reality experiments or experiences that used redirected walking?

Mark only one oval.

- Yes
- No

6. How much previous experience have you had with virtual reality?

Mark only one oval.

0 1 2 3 4 5 6

No experience Very much experience

Qualitative Feedback (Optional)

7. Did you notice any bugs or glitches throughout your playsession?

8. Was the experience enjoyable?

9. How did the redirection techniques feel to you?

10. Did you have any problems throughout the experiment?

11. Is there any way the experience could be improved?

12. Any additional comments?

D Information Sheets

This appendix includes the information and consent sheets that were given to participants throughout Experiment 1 and 2. The sheets are mostly the same with some minor differences in terms of mentioned experiment duration and similar smaller details.

Are you interested in taking part in the research project ”Measuring the quality of redirected walking experiences in virtual reality”?

This is an inquiry about participation in a research project where the main purpose is to measure various quality metrics of state-of-the-art redirection techniques in virtual reality. In this letter I will give you information about the purpose of the project and what your participation will involve.

Purpose of the project

Redirected walking is a field of research in virtual reality where we can make better use of the available physical space by doing small and unnoticeable manipulations to the user as they walk around in a virtual space.

The purpose of this project is to provide data on various quality metrics for state-of-the-art redirection techniques. These metrics include:

- The effect of art style on redirection methods.
- How users move around in the physical space while being redirected.
- How noticeable “distractors” are, which is a redirection technique that allows for higher redirection when the user is engaged with a task or activity.
- The effectiveness of the employed redirection techniques.
- How applicable redirected walking is to games in virtual reality. This is primarily measured through a playtesting session with participants.

This project is a master’s thesis at NTNU Gjøvik.

Who is responsible for the research project?

NTNU Gjøvik is the institution responsible for the project.

Why are you being asked to participate?

You are being asked to participate in this project experiment as you might be part of its target demographic (consisting of young adults or adults with interest in virtual reality). For this experiment, a sample of 15-30 participants is necessary.

What does participation involve for you?

Taking part in this experiment will involve two things:

- Playing through a small virtual reality game in order to provide some data on the various quality metrics. This might consist of multiple play sessions for a total time of approximately 30 minutes.
- Answering a short paper-based survey to provide demographical data and if wanted, some optional qualitative feedback.

The collected data in this experiment consists of:

- Recorded performance and quality metrics from the software side.
- Demographical data like age, gender, previous experience with virtual reality, whether you have taken part in any redirected walking experiments before and if you have needed to remove any optical corrections when participating in the experiment.
- Some qualitative feedback on the experience. Providing this data is optional.

Participation is voluntary

Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. Furthermore, if you at any time start to feel cybersick, uncomfortable or nauseous, you are recommended to stop the experiment. There will be no negative consequences for you if you chose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data

We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act).

The only one in connection to NTNU who will have access to this data is the student responsible for the project (Andreas Wang). The recorded quality metrics and demographical data will be anonymous and only linked through a generated ID number. The performance data will be stored on a publicly available GitHub repository as it cannot be tied to a person without access to the rest of the data. The demographical data will be stored separately in a secure locked box at campus. Similarly, the signature for consent will also be stored separately from the rest of the data in a secure locked box.

No participants will be recognisable in the published results of the master's thesis.

What will happen to your personal data at the end of the research project?

The project is scheduled to end on 01.06.2019. At this point, all the demographical data will be destroyed. The software recorded quality metrics will be archived for future researchers. Since all the demographical data will be destroyed, there is no way to tie these metrics back to any individuals.

Your rights

So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you
- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?

We will process your personal data based on your consent.

Based on an agreement with NTNU, NSD – The Norwegian Centre for Research Data AS has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?

If you have questions about the project, or want to exercise your rights, contact:

- NTNU via supervisors: Simon McCallum(simon.mccallum@ntnu.no) or Christopher Frantz(christopher.frantz@ntnu.no).
- NTNU via student responsible for master's thesis: Andreas Wang(andrwan@stud.ntnu.no).
- Our Data Protection Officer: Thomas Helgesen(thomas.helgesen@ntnu.no).
- NSD – The Norwegian Centre for Research Data AS, by email: (personvertjenester@nsd.no) or by telephone: +47 55 58 21 17.

Yours sincerely,

Andreas Wang

Consent form

Consent can be given in writing (including electronically) or orally. NB! You must be able to document/demonstrate that you have given information and gained consent from project participants i.e. from the people whose personal data you will be processing (data subjects). As a rule, we recommend written information and written consent.

- *For written consent on paper you can use this template*
- *For written consent which is collected electronically, you must chose a procedure that will allow you to demonstrate that you have gained explicit consent (read more on our website)*
- *If the context dictates that you should give oral information and gain oral consent (e.g. for research in oral cultures or with people who are illiterate) we recommend that you make a sound recording of the information and consent.*

If a parent/guardian will give consent on behalf of their child or someone without the capacity to consent, you must adjust this information accordingly. Remember that the name of the participant must be included.

Adjust the checkboxes in accordance with participation in your project. It is possible to use bullet points instead of checkboxes. However, if you intend to process special categories of personal data (sensitive personal data) and/or one of the last four points in the list below is applicable to your project, we recommend that you use checkboxes. This because of the requirement of explicit consent.

I have received and understood information about the project and have been given the opportunity to ask questions. I give consent:

to participate in the experiment.

I give consent for my personal data to be processed until the end date of the project, approx.
01.06.2019

(Signed by participant, date)

Are you interested in taking part in the research project ”Measuring the quality of redirected walking experiences in virtual reality”?

This is an inquiry about participation in a research project where the main purpose is to measure various quality metrics of state-of-the-art redirection techniques in virtual reality. In this letter I will give you information about the purpose of the project and what your participation will involve.

Purpose of the project

Redirected walking is a field of research in virtual reality where we can make better use of the available physical space by doing small and unnoticeable manipulations to the user as they walk around in a virtual space.

The purpose of this project is to provide data on various quality metrics for state-of-the-art redirection techniques. These metrics include:

- The effect of art style on redirection methods.
- How users move around in the physical space while being redirected.
- How noticeable “distractors” are, which is a redirection technique that allows for higher redirection when the user is engaged with a task or activity.
- The effectiveness of the employed redirection techniques.
- How applicable redirected walking is to games in virtual reality. This is primarily measured through a playtesting session with participants.

This project is a master’s thesis at NTNU Gjøvik.

Who is responsible for the research project?

NTNU Gjøvik is the institution responsible for the project.

Why are you being asked to participate?

You are being asked to participate in this project experiment as you might be part of its target demographic (consisting of young adults or adults with interest in virtual reality). For this experiment, a sample of 20-40 participants is estimated to be used.

What does participation involve for you?

Taking part in this experiment will involve two things:

- Playing through a small virtual reality game in order to provide some data on the various quality metrics. This play session is expected to last approximately 10 minutes.
- Answering a short paper-based survey to provide demographical data and if wanted, some optional qualitative feedback.

The collected data in this experiment consists of:

- Recorded performance and quality metrics from the software side.
- Demographical data like age, gender, previous experience with virtual reality, whether you have taken part in any redirected walking experiments before and if you have needed to remove any optical corrections when participating in the experiment.
- Some qualitative feedback on the experience. Providing this data is optional.

Participation is voluntary

Participation in the project is voluntary. If you chose to participate, you can withdraw your consent at any time without giving a reason. All information about you will then be made anonymous. Furthermore, if you at any time start to feel cybersick, uncomfortable or nauseous, you are recommended to stop the experiment. There will be no negative consequences for you if you chose not to participate or later decide to withdraw.

Your personal privacy – how we will store and use your personal data

We will only use your personal data for the purpose(s) specified in this information letter. We will process your personal data confidentially and in accordance with data protection legislation (the General Data Protection Regulation and Personal Data Act).

The only one in connection to NTNU who will have access to this data is the student responsible for the project (Andreas Wang). The recorded quality metrics and demographical data will be anonymous and only linked through a generated ID number. The performance data will be stored on a publicly available GitHub repository as it cannot be tied to a person without access to the rest of the data. The demographical data will be stored separately in a secure locked box at campus. Similarly, the signature for consent will also be stored separately from the rest of the data in a secure locked box.

No participants will be recognisable in the published results of the master's thesis.

What will happen to your personal data at the end of the research project?

The project is scheduled to end on 01.06.2019. At this point, all the demographical data will be destroyed. The software recorded quality metrics will be archived for future researchers. Since all the demographical data will be destroyed, there is no way to tie these metrics back to any individuals.

Your rights

So long as you can be identified in the collected data, you have the right to:

- access the personal data that is being processed about you
- request that your personal data is deleted
- request that incorrect personal data about you is corrected/rectified
- receive a copy of your personal data (data portability), and
- send a complaint to the Data Protection Officer or The Norwegian Data Protection Authority regarding the processing of your personal data

What gives us the right to process your personal data?

We will process your personal data based on your consent.

Based on an agreement with NTNU, NSD – The Norwegian Centre for Research Data AS has assessed that the processing of personal data in this project is in accordance with data protection legislation.

Where can I find out more?

If you have questions about the project, or want to exercise your rights, contact:

- NTNU via supervisors: Simon McCallum(simon.mccallum@ntnu.no) or Christopher Frantz(christopher.frantz@ntnu.no).
- NTNU via student responsible for master's thesis: Andreas Wang(andrwan@stud.ntnu.no).
- Our Data Protection Officer: Thomas Helgesen(thomas.helgesen@ntnu.no).
- NSD – The Norwegian Centre for Research Data AS, by email: (personvertjenester@nsd.no) or by telephone: +47 55 58 21 17.

Yours sincerely,

Andreas Wang

Consent form

Consent can be given in writing (including electronically) or orally. NB! You must be able to document/demonstrate that you have given information and gained consent from project participants i.e. from the people whose personal data you will be processing (data subjects). As a rule, we recommend written information and written consent.

- *For written consent on paper you can use this template*
- *For written consent which is collected electronically, you must chose a procedure that will allow you to demonstrate that you have gained explicit consent (read more on our website)*
- *If the context dictates that you should give oral information and gain oral consent (e.g. for research in oral cultures or with people who are illiterate) we recommend that you make a sound recording of the information and consent.*

If a parent/guardian will give consent on behalf of their child or someone without the capacity to consent, you must adjust this information accordingly. Remember that the name of the participant must be included.

Adjust the checkboxes in accordance with participation in your project. It is possible to use bullet points instead of checkboxes. However, if you intend to process special categories of personal data (sensitive personal data) and/or one of the last four points in the list below is applicable to your project, we recommend that you use checkboxes. This because of the requirement of explicit consent.

I have received and understood information about the project and have been given the opportunity to ask questions. I give consent:

to participate in the experiment.

I give consent for my personal data to be processed until the end date of the project, approx.
01.06.2019

(Signed by participant, date)

E Approval - NSD

The following two pages consist of a copy of NSD's approval for the data collection in this thesis.

NSD NORSK SENTER FOR FORSKNINGSDATA

NSD sin vurdering

Prosjekttittel

Noticeability and Effectiveness of Distractors in Redirected Walking

Referansenummer

404493

Registrert

07.02.2019 av Andreas Wang - andrwan@stud.ntnu.no

Behandlingsansvarlig institusjon

NTNU Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for datateknologi og informatikk

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)

Simon McCallum, simon.mccallum@ntnu.no, tlf: 64225019481

Type prosjekt

Studentprosjekt, masterstudium

Kontaktinformasjon, student

Andreas Wang, andrwan@stud.ntnu.no, tlf: 48048162

Prosjektperiode

07.01.2019 - 01.06.2019

Status

10.02.2019 - Vurdert

Vurdering (1)

10.02.2019 - Vurdert

Our assessment is that the processing of personal data in this project will comply with data protection legislation, presupposing that it is carried out in accordance with the information given in the Notification Form and attachments, 10.02.2019. Everything is in place for the processing to begin.

NOTIFY CHANGES

If you intend to make changes to the processing of personal data in this project it may be necessary to notify NSD. This is done by updating the information registered in the Notification Form. On our website we explain which changes must be notified. Wait until you receive an answer from us before you carry out the changes.

TYPE OF DATA AND DURATION

The project will be processing general categories of personal data until 01.06.2019.

LEGAL BASIS

The project will gain consent from data subjects to process their personal data. We find that consent will meet the necessary requirements under art. 4 (11) and 7, in that it will be a freely given, specific, informed and unambiguous statement or action, which will be documented and can be withdrawn. The legal basis for processing personal data is therefore consent given by the data subject, cf. the General Data Protection Regulation art. 6.1 a).

PRINCIPLES RELATING TO PROCESSING PERSONAL DATA

NSD finds that the planned processing of personal data will be in accordance with the principles under the General Data Protection Regulation regarding:

- lawfulness, fairness and transparency (art. 5.1 a), in that data subjects will receive sufficient information about the processing and will give their consent
- purpose limitation (art. 5.1 b), in that personal data will be collected for specified, explicit and legitimate purposes, and will not be processed for new, incompatible purposes
- data minimisation (art. 5.1 c), in that only personal data which are adequate, relevant and necessary for the purpose of the project will be processed
- storage limitation (art. 5.1 e), in that personal data will not be stored for longer than is necessary to fulfil the project's purpose

THE RIGHTS OF DATA SUBJECTS

Data subjects will have the following rights in this project: transparency (art. 12), information (art. 13), access (art. 15), rectification (art. 16), erasure (art. 17), restriction of processing (art. 18), notification (art. 19), data portability (art. 20). These rights apply so long as the data subject can be identified in the collected data.

NSD finds that the information that will be given to data subjects about the processing of their personal data will meet the legal requirements for form and content, cf. art. 12.1 and art. 13.

We remind you that if a data subject contacts you about their rights, the data controller has a duty to reply within a month.

FOLLOW YOUR INSTITUTION'S GUIDELINES

NSD presupposes that the project will meet the requirements of accuracy (art. 5.1 d), integrity and confidentiality (art. 5.1 f) and security (art. 32) when processing personal data.

To ensure that these requirements are met you must follow your institution's internal guidelines and/or consult with your institution (i.e. the institution responsible for the project).

FOLLOW-UP OF THE PROJECT

NSD will follow up at the planned end date in order to determine whether the processing of personal data has been concluded.

Good luck with the project!

Data Protection Services for Research: +47 55 58 21 17 (press 1)

