



**A University of Sussex PhD thesis**

Available online via Sussex Research Online:

<http://sro.sussex.ac.uk/>

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

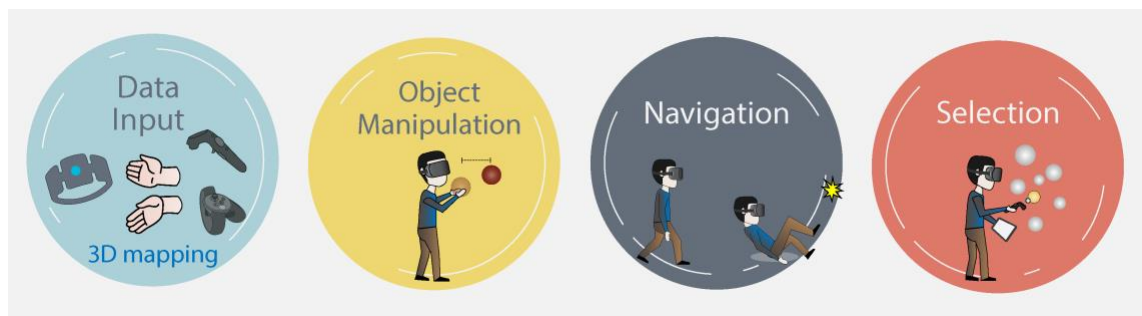
The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details

# Computational Interaction Techniques for 3D Selection, Manipulation and Navigation in Immersive VR

Roberto A. Montano-Murillo



Thesis submitted for the degree of Doctor of Philosophy

School of Engineering and Informatics

University of Sussex

May 2019

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature:

## Summary

3D interaction provides a natural interplay for HCI. Many techniques involving diverse sets of hardware and software components have been proposed, which has generated an explosion of Interaction Techniques (ITes), Interactive Tasks (ITas) and input devices, increasing thus the heterogeneity of tools in 3D User Interfaces (3DUIs). Moreover, most of those techniques are based on general formulations that fail in fully exploiting human capabilities for interaction. This is because while 3D interaction enables naturalness, it also produces complexity and limitations when using 3DUIs.

In this thesis, we aim to generate approaches that better exploit the high potential human capabilities for interaction by combining *human factors*, mathematical formalizations and computational methods. Our approach is focussed on the exploration of the close coupling between specific ITes and ITas while addressing common issues of 3D interactions.

We specifically focused on the stages of interaction within Basic Interaction Tasks (BITas) i.e., *data input*, *manipulation*, *navigation* and *selection*. Common limitations of these tasks are: (1) the complexity of mapping generation for *input* devices, (2) fatigue in mid-air object *manipulation*, (3) space constraints in VR *navigation*; and (4) low accuracy in 3D mid-air *selection*.

Along with two chapters of introduction and background, this thesis presents five main works. Chapter 3 focusses on the design of mid-air gesture mappings based on human tacit knowledge. Chapter 4 presents a solution to address user fatigue in mid-air object *manipulation*. Chapter 5 is focused on addressing space limitations in VR *navigation*. Chapter 6 describes an analysis and a correction method to address *Drift* effects involved in *scale-adaptive* VR *navigation*; and Chapter 7 presents a hybrid technique 3D/2D that allows for precise selection of virtual objects in highly dense environments (e.g., point clouds). Finally, we conclude discussing how the contributions obtained from this exploration, provide techniques and guidelines to design more natural 3DUIs.

## Table of Contents

Summary .....	iii
List of Figures .....	ix
List of Tables .....	xi
List of Abbreviations .....	xii
Chapter 1 Introduction .....	1
1.1 Motivation .....	1
1.2 Opportunities, Goals and Approach .....	3
1.3 Thesis Structure .....	4
Chapter 2 3D User interfaces: Background, Opportunities & Summary of the Approach .....	9
2.1 Elements of a 3D User Interface .....	9
2.1.1 Virtual Space .....	11
2.1.2 Virtual Objects .....	12
2.1.3 Interactive Dialogue .....	14
2.2 Opportunities and Justification of the Approach .....	19
2.2.1 Mapping Generation .....	21
2.2.2 Interaction Techniques (ITes) .....	21
2.2.3 Interactive Tasks (ITas) .....	22
2.2.4 About Isomorphic and Non-isomorphic Mapping .....	22
Chapter 3 Optimization of Text-Entry Devices with Low Gestural Resolution .....	24
3.1 Introduction .....	24
3.2 Related Work .....	27
3.2.1 Gestural Input Devices: A growing landscape .....	27
3.2.2 Gestures and Mappings: Point Studies .....	28
3.3 Our approach: Semi-automatic Mappings for Low Input Resolution .....	29
3.3.1 Quantification of Low-level Factors and Constraints .....	29
3.3.2 Domain Expert Knowledge Elicitation .....	30
3.3.3 Formalization & Optimization .....	30
3.3.4 Comparative-Summative Evaluation .....	31

3.4	Case Study with Myo: Compute Vs Design .....	31
3.4.1	Problem Delimitation .....	32
3.4.2	Quantification of Relevant Factors. ....	33
3.4.3	Designer's Workshop.....	36
3.4.4	Formalization & Optimization. ....	38
3.4.5	Computing alternative approaches.....	40
3.5	Analytical and Summative Evaluation.....	41
3.5.1	Experiment Setup .....	42
3.5.2	Analysis of Results.....	44
3.6	Discussion .....	46
3.7	Conclusion .....	48
Chapter 4 Erg-O: Ergonomic Optimization of Immersive Virtual Environments .....		49
4.1	Introduction .....	49
4.2	Related Work .....	52
4.2.1	Visual Dominance and Spatial Redirecting .....	52
4.2.2	Ergonomics .....	53
4.3	Erg-O: ergonomic Optimization for Redirected Interaction .....	54
4.3.1	Retargeted Manipulation: Bijective Mapping of Visual and Physical Spaces 56	
4.3.2	Retargeted Space Partitioning .....	58
4.3.3	Real-time Update of the Trees. ....	59
4.4	Computing Retargeting Mappings .....	60
4.4.1	Quantifying Retargeted Mappings .....	60
4.4.2	Optimization Methods to Compute Mappings.....	62
4.4.3	Analytical Comparison of Retargeting Approaches .....	63
4.5	User Study.....	65
4.5.1	Task and Environment .....	65
4.5.2	Layouts Tested: .....	65
4.5.3	Experimental Design.....	66
4.6	Results and Analysis .....	67

4.6.1	General Analysis of Retargetting Approaches .....	67
4.7	Discussion .....	71
4.8	Conclusion .....	72
Chapter 5 NaviFields: Relevance fields for adaptive VR navigation .....		73
5.1	Introduction .....	73
5.2	Related Work .....	75
5.2.1	VR Navigation Techniques .....	75
5.2.2	Dynamic Control of the Viewpoint.....	77
5.3	NaviFields: Adaptive VR Navigation .....	78
5.3.1	Modelling Navi-Fields: Differential Tracking .....	79
5.3.2	Generating the Navigation Field $k(MWUt)$ :.....	80
5.4	User Study .....	82
5.4.1	Participants .....	82
5.4.2	Testing Environment and Navigation Tasks .....	83
5.4.3	Experimental Design .....	85
5.4.4	Scaling Factors, Environment Size and Navigation Fields .....	86
5.4.5	Usability Evaluation Criteria .....	87
5.5	Results and Partial Discussion .....	88
5.5.1	Traveling task: NaviFields vs. Homogeneous Scaling .....	88
5.5.2	Manoeuvring: NaviFields vs. Homogeneous Scaling.....	91
5.6	Drifting Effects in NaviFields .....	94
5.7	Discussion .....	95
5.8	Conclusions .....	97
Chapter 6 Drift-Correction Techniques for <i>Scale-Adaptive</i> VR Navigation .....		98
6.1	Introduction .....	98
6.2	Related Work .....	100
6.2.1	VR Navigation Techniques .....	101
6.2.2	<i>Drift</i> Effect in VR Navigation.....	102
6.3	Modelling <i>Scale-adaptive</i> Techniques.....	103
6.3.1	Uncorrected <i>Scale-adaptive</i> Techniques: A General Model.....	103

6.3.2	Translational Gain ( $G(t): \mathbb{R} \rightarrow \mathbb{R}_3$ ): .....	104
6.3.3	Defining <i>Drift</i> : Uncontrolled Gain along Closed Paths .....	105
6.3.4	Estimated <i>Drift</i> : Approximation for Open Paths .....	105
6.4	Study 1. Quantifying and Understanding <i>Drift</i> .....	106
6.4.1	Analysis of <i>Drift</i> Effects .....	107
6.5	<i>Drift</i> Correction Techniques .....	108
6.5.1	A Generic Model for <i>Drift</i> Correction Techniques.....	109
6.5.2	Derivative Correction.....	109
6.5.3	Angular Correction .....	109
6.6	Testing & Tuning our Correction Techniques .....	110
6.6.1	Data Driven Simulation .....	111
6.6.2	Simulation of Correction Configurations and Results .....	111
6.7	Study 2. Generability and Robustness of our Correction Techniques .....	113
6.7.1	Seven League Boots Implementation (7LB).....	114
6.7.2	Environment and NaviFields Implementation (NF) .....	114
6.7.3	Navigation Tasks.....	115
6.7.4	Experimental Design and Variables Measured .....	116
6.7.5	Analysis of Results.....	117
6.8	Discussion .....	120
6.9	Conclusion .....	121
Chapter 7 Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments .....		123
7.1	Introduction .....	123
7.2	Related work .....	127
7.2.1	Occlusion in 2D/3D Selection.....	127
7.2.2	Selection using Mobile Devices in Mixed Reality.....	128
7.3	Implementation and Setup .....	130
7.3.1	Slicing Volume .....	130
7.3.2	3D-2D Mapping (Mid-air & Tablet) .....	132
7.3.3	Formal Description .....	133



7.4	User Study .....	137
7.4.1	Participants .....	137
7.4.2	Task Point Cloud Model .....	137
7.4.3	Experimental Conditions.....	139
7.4.4	Experimental Setup .....	141
7.4.5	Variables measured .....	141
7.4.6	Results .....	142
7.5	Discussion .....	146
7.6	Limitations and future work.....	148
7.7	Conclusion .....	149
	Chapter 8 Conclusions .....	150
8.1	Individual Chapter Contributions.....	151
8.2	Interaction Stages & General Contributions .....	153
8.3	Conclusion .....	156
	References .....	157
	Appendix 1 .....	174
	Appendix 2 .....	176
	Appendix 3 .....	188

## List of Figures

Figure 1.1 (left) Sensory homunculus .....	2
Figure 2.1 Diagrams of 3DUI components .....	10
Figure 2.2 <i>Virtual space</i> components .....	11
Figure 2.3 Virtual Object components .....	12
Figure 2.4 Model proposed by Molina to describe the ITes' stages .....	17
Figure 2.5 Diagram to illustrate our aim to better exploit <i>human factors</i> .....	20
Figure 3.1 Resulting mappings from the full optimization .....	26
Figure 3.2 Gestures possible with a Myo armband.....	32
Figure 3.3 Two-step chain gestures under designers' categories.....	32
Figure 3.4 Mappings' performance.....	34
Figure 3.5 Average time (a) and accuracy (b).....	35
Figure 3.6 Interface layout .....	37
Figure 3.7 Final gesture mappings .....	43
Figure 3.8 Experimental setup for the typing task. ....	43
Figure 3.9 Comparison of performance .....	45
Figure 3.10 User Experience results .....	45
Figure 3.11 Percentage of participants that preferred each mapping.....	46
Figure 4.1 Erg-O approach.....	50
Figure 4.2 RULA comfort mapping representation .....	54
Figure 4.3 Example of a tetrahedron pair.....	57
Figure 4.4 Summary of our manipulation technique .....	58
Figure 4.5 User displacements .....	59
Figure 4.6 Comparison of the retargeting behaviour .....	62
Figure 4.7 Experimental task .....	64
Figure 4.8 Box plots for the results of our questionnaires .....	68
Figure 4.9 Average measurements per strategy and layout .....	70
Figure 5.1 NaviFields approach .....	74
Figure 5.2 NaviFields' principle .....	80
Figure 5.3 Modelling navigation fields.....	81
Figure 5.4 Virtual environment.....	83
Figure 5.5 ParaFrustums .....	84
Figure 5.6 Results of traveling task .....	89
Figure 5.7 Results of manoeuvring task.....	93
Figure 5.8 Example of the role of <i>Drift</i> .....	94
Figure 6.1 The <i>Drift</i> effect .....	99
Figure 6.2 Illustration of user displacement.....	104
Figure 6.3 Drifted trajectory .....	104

Figure 6.4 Travelling task .....	107
Figure 6.5 Evolution of the average of estimated <i>Drift</i> .....	108
Figure 6.6 Overview of the results (data driven simulation) .....	112
Figure 6.7 Comparison of <i>Drift</i> ( $\alpha=0.25$ , $\epsilon=0.25$ , best configuration) .....	112
Figure 6.8 Test environment and <i>navigation field</i> used for the second study.....	115
Figure 6.9 <i>Drift</i> in NaviFields .....	117
Figure 6.10 <i>Drift</i> in Seven League Boots .....	117
Figure 6.11 NaviFields: Comparison of performance (NF Vs <i>NFC</i> ).....	118
Figure 6.12 Seven League Boots: Comparison of performance (7LB Vs <i>7LBC</i> ).....	118
Figure 6.13 Questionnaire results for NF and <i>NFC</i> . .....	119
Figure 6.14 Questionnaire results for 7LB and <i>7LBC</i> . .....	119
Figure 7.1 Slicing volume 3D-2D interaction.....	125
Figure 7.2 Propagation technique .....	126
Figure 7.3 Pen-and-tablet metaphor .....	131
Figure 7.4 System node graph.....	133
Figure 7.5 The general behaviour of the system. ....	135
Figure 7.6 Virtual Environment used in the user study .....	138
Figure 7.7 Controller actions for the three interaction techniques.....	139
Figure 7.8 Selection modalities.....	140
Figure 7.9 Experimental setup. ....	141
Figure 7.10 Results on Accuracy .....	143
Figure 7.11 Results on Over-Selection .....	143
Figure 7.12 Results on Time .....	144
Figure 7.13 Results on user experience.....	145
Figure 7.14 Participants' preferred interaction technique.....	145
Figure S1. Mean <i>Drift</i> uncorrected (red bars) and resultant <i>Drift</i> .....	189
Figure S2. Mean of real and virtual distances travelled.....	192
Figure S3. Mean of real and virtual distances per flag travelled .....	192

## List of Tables

Table 2.1 Virtual object types .....	13
Table 2.2 Foley's analogy between interactive dialogue & the elements of a sentence .	14
Table 2.3 Basic Interactive Tasks (BITas).....	16
Table 3.1 Factors used use for <b><i>MDes</i></b> .....	40
Table 3.2 Numerical mappings' performance.....	42
Table 5.1 Results from repeated measures ANOVA .....	89
Table 5.2 ANOVA effects for each independent variable. ....	91
Table 5.3 ANOVA effects for the interactions of the factor T .....	91
Table 5.4 ANOVA effects for questionnaire questions and interactions of factor T .....	91
Table 6.1 Variables measured to assess technique performance. ....	116
Table S1. Mean <i>Drift</i> of each technique (Uncorrected, <i>Derivative</i> and <i>Angular</i> ) .....	188
Table S2. Summary of <i>Drift</i> effects in each correction technique .....	190
Table S3. Summary of effects of the correction techniques on performance. ....	190
Table S4. List of statements used in our subjective evaluation. ....	191
Table S5. Summary of effects of the correction techniques .....	191
Table S6. Real and virtual distances travelled .....	192

## List of Abbreviations

Abbreviations	Meaning
2D	Two-Dimensional
3D	Three-Dimensional
VR	Virtual Reality
IVR	Immersive Virtual Reality
VE	Virtual Environment
CVE	Collaborative Virtual Environment
AR	Augmented Reality
UI	User Interface
3DUI	Three-Dimensional User Interface
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HMD	Head Mounted Display
ITe	Interaction Technique
ITa	Interactive Task
BITa	Basic Interactive Task
CITa	Composite Interactive Task
DoF	Degrees of Freedom
UX	User Experience
IMU	Inertial Measurement Unit
EMG	Electromyography
WIMP	Window-Icon-Mouse-Pointer
RULA	Rapid Upper Limb Assessment
JND	Just Noticeable Threshold
IK	Inverse Kinematics
WIP	Walking In Place technique
fps	Frames Per Second
FoV	Field of View
7LB	Seven league Boots technique
MRI	Magnetic Resonance Imaging
CT	Computerized Tomography
BCI	Brain Computer Interface
HF	Human Factors
RT	Real Time

# Chapter 1

## Introduction

### 1.1 Motivation

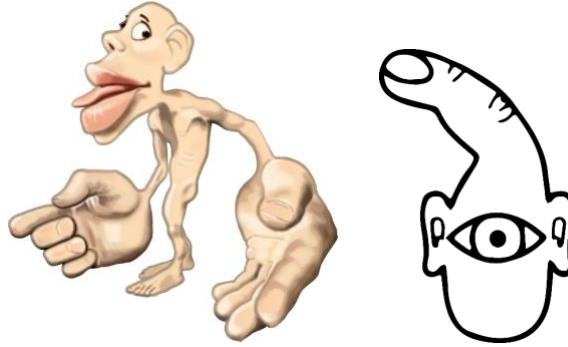
The usability requirements for designing User Interfaces (UIs) are well known and commonly standardized for 2D applications ([LaViola Jr et al., 2017](#)). For instance, the WIMP metaphor ([Van Dam, 1997](#)) used in 2D User Interfaces (2DUIs), has been widely accepted, and a standardized set of Interaction Techniques (ITes), Basic Interactive Tasks (BITas) and controls have been identified to such an extent, that they are natively supported by all modern operating systems (OS). However, while 2DUIs are widely used, they do not fully exploit common human capabilities to interact with the real world.

According to neurological maps of human body representation, humans have large sensory capabilities as illustrated in the *sensory homunculus* (**Figure 1.1** left). This representation shows scaled proportions of the body according to the amount of cortex in the brain devoted to sensory functions e.g., large hands representing high touch sensibility and high precision of movements ([Penfield et al., 1937](#); [Penfield et al., 1950](#)).

On the other hand, our interaction with traditional 2DUIs is limited. According to O’Sullivan and Igoe, the way “*how the computer sees us*” is a representation consisting of only one eye (as we don’t need any depth), one finger (due to common tapping input with the same finger) and two small ears (to perceive audio feedback) as shown in **Figure 1.1** right. ([O’Sullivan et al., 2004](#)).

Even considering the increasing emergence of multitouch input devices (involving multiple fingers) and mobile computing that allows users’ navigation (involving actual walking), the lack of 3D worlds hinders richness and naturalness in interaction.

By comparing the two representations depicted in **Figure 1.1**, it is clear that human capabilities are not fully exploited when using 2DUIs, suggesting in turn a lack of naturalness as 2D interaction does not mimic the way we actually interact with real 3D worlds. Thus, “*to change how the computer reacts to us, we have to change how it sees us*” (O'Sullivan et al., 2004).



**Figure 1.1 (left) Sensory homunculus illustrating human capabilities to interact with real 3D worlds. Figure source: (Brown University, 2014). (Right) Illustration of “how the computer sees us” when interacting with 2DUIs (O'Sullivan et al., 2004).**

3D User Interfaces (3DUIs) arguably offer the highest potential to exploit human capabilities. For instance, Virtual Reality (VR) stimulates all of the user senses, simulating thus the way we explore and interact in the real world (Sanchez-Vives et al., 2005). Indeed, this technology has become mainstream due to many applications over a wide range of fields e.g., medicine, education, teleoperation, training and entertainment (LaViola Jr et al., 2017). However, the potential of VR comes at the expense of much higher complexity.

For instance, to unleash human interaction capabilities, VR relies on a wide range of heterogeneous hardware (e.g., body motion sensing, eye tracking and electromyography (EMG), etc.). Successively, while VR offers natural interplay by means of body tracking and gestural input, mid-air interaction also involves more complex tasks e.g., “*touching a menu item floating in space is much more difficult than selecting a menu item on the desktop, not only because the task has become 3-D, but also because the important constraint of the physical desk on which the mouse rests is missing.*” (Bowman et al., 2001). Therefore, even when VR provides richness of interaction in 3D worlds, its complexity makes the standardization across a variety of ITes, BITas and controls even more difficult.

Certainly, important improvements have been achieved along the years, especially in terms of sensing approaches (relying on optical systems, EMG; radar and sonar

technologies) to such an extent that consumer-grade devices are currently available to the public at affordable prices (e.g., Kinect and Leap Motion). This has enabled attempts of standardization of 3DUIs (e.g., Google and Microsoft), one recent example is Unity, which supports most of the consumer-grade VR devices and OpenVR provides the promise of standardized support to VR headsets, controllers and tracking systems. Yet, this is still hindered by the broad spectrum of possibilities that VR enables. This suggests that advances in hardware have addressed some of the issues related to 3DUIs (e.g., in terms of perception, rendering, frame rate, weight and wires), but these solutions are still generic i.e., they do not exploit the interplay between the specific task the user needs to complete, and the potential of human interaction/perception.

As a consequence of this increased complexity and the heterogeneity of hardware, no standard set of ITes has been developed. Despite some conceptual frameworks being available (e.g., virtual hand, ray-casting, etc.) their solutions are also generic and revolve around general metaphors. Therefore, there are no development frameworks comparable to those available for standardized 2DUIs ([Dachselt et al., 2005](#)).

## 1.2 Opportunities, Goals and Approach

In this thesis, we take advantage of both (1) the high potential of 3DUIs to exploit human capabilities for interaction and (2) the hardware improvements that have broadened the scope of applications.

However, fully exploiting these two benefits in 3DUIs, requires a close coupling between the specific scope of the application (i.e., the task the user needs to complete) and the ITes used, e.g., pointing (the ITa), achieved by ray-casting technique (the ITe). As general 3DUI metaphors were proposed many years ago, they fail in exploiting the potential interplay between the *human factors* and the specific task that the specific 3DUI is designed for. First, being general metaphors, they have very limited ability to be tailored for specific task. Second, being proposed many years ago, they were not designed with the computational power of current devices in mind, and rely on relatively simple algorithms.

Our goal is therefore, to help on the generation of more natural 3DUIs by exploiting the close coupling between *human factors* and ITas (tasks). We achieve this by employing mathematical formalizations and computational approaches, taking advantage of the computational power of current devices.



### 1.3 Thesis Structure

The very own scope of 3DUIs, dealing with heterogeneous hardware, complex interaction, huge range of ITas and non-standard ITes, makes it hard to provide a comprehensible exploration of our goal with the scope of a thesis. Thus, this thesis explores the potential of coupling *human factors* and ITes by illustrating applications of our approach to each of the main stages of interaction in 3DUIs i.e., mappings, ITes and ITas. Particularly, the exploration performed within this thesis is structured as follow:

First, [Chapter 2](#) provides background related to the current state of 3DUIs when compared with traditional GUIs, identifying possible improvement opportunities. This chapter also details our main approach and how each point study fits in the general scope of 3DUIs.

In [Chapter 3](#), we explore the problem of mapping complexity on hardware inputs. *Data input* is the first stage of the *interactive dialogue* ([Foley et al., 1990](#)), and one of the most important elements in any VR interaction technique as it is the link between real and virtual worlds ([LaViola Jr et al., 2017](#)). One of the most intuitive metaphors for 3D *data input* is gestural interaction ([Quek, 1996](#)). From early projects like Krueger’s Videoplace ([Krueger et al., 1985](#)), developers have been fascinated by using hand motion as direct input. However, designing gesture mappings for 3D interaction is complicated ([LaViola, 2013](#)). Particularly challenging is the elicitation of the most appropriate gestures and their mapping to tasks, which can easily lead to a combinatorial explosion. This is due to the high resolution of human hand motion, with many degrees of freedom “*the hand is the most dexterous of the extremities*” ([Jones et al., 2006](#); [Sridhar et al., 2015](#)). As a result, interaction designers are faced with a very challenging task, with many factors involved in the creation of the gestural interfaces.

To address this issue, this chapter illustrates a case of study of low gestural resolution input devices (e.g., Myo Armband) used to achieve a “complex *interactive dialogue*” with a task that requires a high input resolution (text entry). We present an approach for the semi-automatic generation of gesture mappings in a text entry task. Our solution was based on the optimization of mathematically modelled factors collected from systematic studies (*-low level factors-* i.e., error, speed, accuracy) and semi-structured workshops with experts (*-high level factors-* e.g., cognitive load, heuristics). The comparison between our approach (combining *high-level* and *low-level* factors) with other computationally determined mappings using naive cost functions (*low-level* factors only) showed that our mapping presented a good balance on performance in all factors involved

(speed, accuracy, comfort, memorability, etc.), consistently performing better than purely computational mappings. The results indicate that our hybrid approach can yield better results compared to a pure user-driven methodology or pure computational approaches and show an interesting interplay between *human factors* and computational approaches. This work was published in INTERACT 2019. DOI: [10.1007/978-3-030-29381-9\\_38](https://doi.org/10.1007/978-3-030-29381-9_38)

Then, in [Chapter 4](#) we explore 3D *Object Manipulation*, one of the major group of interaction techniques in the second stage of the *interactive dialogue* i.e., “*manipulations that preserve the shape of objects*” (Foley et al., 1984; LaViola Jr et al., 2017). Virtual 3D *manipulation* should be analogous to the way we perform it in the real world (e.g., reaching and grabbing objects). Many parameters are considered to achieve this analogous interaction, for example “distance to the target object, target size; the density of objects around the target” (Poupyrev et al., 1997). Particularly, distance is often a limitation in terms of ergonomics, as *manipulation* within certain distances is highly constrained by interaction within arms’ reach, task space and tracking devices (Mine, 1995). Since 3DUIs usually require mid-air gestures and hand-coupled devices, *object manipulation* is subject to fatigue and discomfort due to the height and distance of the hands with respect to the body, producing thus the so called “gorilla arm effect” (Boring et al., 2009). This effect increases when uncomfortable and unsupported poses are performed for long periods of time i.e., “the farther away and higher the hands, the less time users can interact comfortably” (LaViola Jr et al., 2017).

To address this issue, in this chapter we then present an *object manipulation* technique (that we called Erg-O), that increases the *endurance time* i.e., the duration of user comfort (Chaffin et al., 1999). This technique leverages visual dominance to maintain the visual location of the elements in VR, while making them accessible from more comfortable physical locations (using a retargeting approach). We explore minimum redirections, producing a non-isomorphic mapping of the visual and physical spaces to improve comfort, but allowing for non-noticeable redirection to retain naturalness. We used an ergonomic evaluation combined with real-time optimization approaches to compute the best physical location to interact with each visual element (comfort mapping), and space partitioning techniques to distort the visual and physical spaces based on the generated comfort mapping to allow multi-object retargeting. Our studies tested our approach, comparing two different retargeting strategies using Erg-O versus a traditional virtual hand method. Our results demonstrated the potential of our technique to improve

ergonomics, without significant effects on performance. This example illustrates the possibility to use real-time optimization approaches to improve *human factors* (i.e., ergonomics), while not hindering usability criteria (e.g., sense of control, performance). This work was published in UIST 2017. DOI: [10.1145/3126594.3126605](https://doi.org/10.1145/3126594.3126605).

[Chapter 5](#) then explores *Navigation* techniques i.e., the second major group of interaction techniques, again situated in the second stage of the *interactive dialogue*. In 3DUIs, *travel* achieved by actual physical motion such as walking (i.e., when body tracking is used) is only adequate within constrained space and speed ([LaViola Jr et al., 2017](#)). This is due to limitations of physical space (room size) and tracking ranges. For this reason, additional controls are employed e.g., vehicle metaphor and teleportation ([Bowman et al., 2004](#)). However, these techniques usually involve only visual motion cues limiting vestibular cues, producing thus cybersickness and disorientation ([Bowman et al., 1997](#)). Therefore, adding vestibular cues that provide walking-like perception, is desirable to avoid side effects ([Harris et al., 1999](#)) caused by visual-vestibular mismatch.

In this chapter, we particularly focus on *scaled-adaptive* navigation techniques, as they allow “full gait” metaphor ([LaViola Jr et al., 2017](#)) by using a non-isomorphic mapping between real and virtual displacements, i.e., dynamically scaling the user’s movements ([Interrante et al., 2007](#)) enabling user’s *travel* through Virtual Environments (VEs) larger than the physical space. However, although this technique provides vestibular cues and expands the navigable space, it is often considered unnatural ([Steinicke et al., 2010](#)) because scaling factors applied are only useful for large displacements along non-relevant areas. For fine *navigation* tasks however (e.g., search and *manoeuvring*), it is not suitable. This is because *scaled-navigation* is usually based on egocentric control (when walking view is fully controlled by the user) and it is not space-aware ([Thorndyke et al., 1982](#)).

To address this issue, in this chapter we propose a novel method (that we called NaviFields) that employs *scale-adaptive* VR *navigation*, which quantifies the requirements for precise *navigation* of each point of the environment, allowing natural *navigation* (1:1 mapping) within relevant areas, while scaling users’ displacements when travelling across non-relevant spaces. This expands the size of the navigable space, retains the natural *navigation* metaphor and still allows for areas with precise control of the virtual head. Results from a user evaluation showed NaviFields’ ability to cover larger spaces, introduce minimal disruption when travelling across bigger distances and improve significantly the precise viewpoint control inside relevant areas. This shows an

application of our approach enabling locomotion via physical displacement (i.e., exploit *human factors*), not hindering usability (i.e., performance was similar to natural walk in many cases) and alleviating constraints within 3DUIs (i.e., enable walking along larger spaces). This work was published in UIST 2017. DOI: [10.1145/3126594.3126645](https://doi.org/10.1145/3126594.3126645).

Subsequently, [Chapter 6](#) highlights a limitation found in the previous chapter. That is, the technique presented in [Chapter 5](#) showed great potential, but it also introduced a side effect (*Drift*) resulting from gradually scaling displacements as the user moves causing a mismatch between the real/virtual spaces that can grow over time, and turn the *navigation* techniques unusable. This *Drift* seriously limited the amount of time that a user could make use of our technique. Interestingly, the same issue affected other *scale-adaptive VR navigation* techniques. To address this issue, in this chapter we present an analysis of the effects of *Drift*, highlighting its potential detrimental effects. Then, two techniques to correct *Drift* effects are described and applied to two different *scale-adaptive navigation* techniques: one *ego-centric* (Seven League Boots) and one *allocentric* (NaviFields), demonstrating that our correction technique can significantly reduce *Drift* effects, extend the life-span of the *navigation* techniques (i.e., time that they can be used before *Drift* draws targets unreachable), while not affecting users' experience and not adding any significant changes to the basic behaviour of the technique. This chapter is not explicitly focussed on enabling *human factors*, although enabling the use of physical displacements for longer periods of time could be considered under this light. Even if focussed on fixing a technical issue inherited from [Chapter 5](#), this chapter illustrates the value of formal mathematical modelling (*Drift*) and the use of real-time simulations to optimize displacements. This work was published in UIST 2019. DOI: [10.1145/3332165.3347914](https://doi.org/10.1145/3332165.3347914).

In [Chapter 7](#) we then explored 3D *selection* tasks i.e., one of the most important interaction techniques situated in the third stage of the *interactive dialogue*. Object *selection* has been identified as a fundamental task in 3DUIs ([Mine, 1995](#); [LaViola Jr et al., 2017](#)) and one of the primary task in VR interaction ([Bowman et al., 1999](#)). One significant limitation that affects most of the 3D *selection* techniques is inaccuracy due to inter-object distance, scene-density and occlusion ([LaViola Jr et al., 2017](#)). Unlike 2D-based *selection* metaphors (e.g., mouse and touchscreens), 3D *selection* suffers from more salient effects of inaccuracy as it often involves mid-air interaction in 3D unconstrained spaces (e.g., 3D drawing in VR ([Arora et al., 2018](#))) and floating menus ([Bowman et al., 2001](#))). Therefore, tabletop-based 2D interaction techniques have been proposed as they

allow well known 2D metaphors in 3D *selection* tasks, taking advantage of accuracy and stability of having a surface (e.g., “Balloon Selection” (Benko et al., 2007), “Void Shadows” (Giesler et al., 2014) and “Triangle Cursor” (Strothoff et al., 2011)).

However, *selection* is still challenging in 3D world coordinates where the main interplay requires full mid-air interaction (e.g., in Immersive Virtual Reality - IVR) and Head Mounted Displays (HMD) (unlike tabletop surfaces) for both *selection* and *manipulation* tasks. This issue is specially limiting in dense environments (e.g., point cloud editing). Today, the availability of 3D scanners and imaging (e.g., photometric scanning (Izadi et al., 2011) and 3D MRI and CT scans (Altahawi et al., 2018)) allow artists and scientists to obtain virtual representations of the real world in form of large data set of points. While VR can provide an intuitive medium to interact with these 3D models, mid-air interaction and high density are still limitations for analysing, cleaning and editing 3D data.

To address this limitation, in this chapter we propose “Slicing-Volume”, a hybrid *selection* technique that enables simultaneous 3D interaction in mid-air and a 2D pen-and-tablet metaphor in VR. Inspired by well-known slicing plane techniques in data visualization, our approach consists of a 3D volume that encloses target objects in mid-air, which are then mapped to a 2D tablet view for precise *selection* on a tangible physical surface. We evaluated the accuracy of our approach in a highly occluded *selection* task, comparing different multimodal interaction techniques (e.g., mid-air, virtual tablet and real tablet). Our results showed that our hybrid technique significantly improved accuracy of *selection* thanks to the added visualization and the combined benefits of mid-air and 2D tablet interactions. Our system then provides a solution for point cloud editing tasks suffering from high occlusion levels. In line with our main goal, this chapter illustrates how we exploit *human factors* enabling bimanual interaction for more precise object *selection*, using hybrid strategies in VR.

Finally, in Chapter 8 we reflect on the results obtained during each of the *interactive dialogue* stages explored in this work (mapping, ITes and ITas), as well as the general overview about the potential of the main approach in this thesis. We also discuss our contributions related to the solutions provided in each chapter, along with possible scenarios where our findings can benefit the integration of *human factors* into the 3DUIs design. We highlight how our general approach can help in gaining naturalness on 3D interactions, improving thus the generation of tailored user interfaces that better reflect the human allowances and limitations in terms of interaction.

## Chapter 2

# 3D User Interfaces: Background, Opportunities & Summary of the Approach

In this chapter, we review important concepts, components, and classifications of 3DUIs. This review aimed to ground the basis where this thesis is built on, enabling to analyse the 3DUI components independently to better identify relevant issues and propose solutions not only from the VR point of view, but also from different levels of abstraction that BITas classification and the *interactive dialogue* involve. This topic's segmentation will allow us to better recognize potential solutions where *human factors* can be used to enhance the interaction with 3DUIs, thus exploiting human capabilities to generate more natural an intuitive interaction.

### 2.1 Elements of a 3D User Interface

Hix and Hartson defined a UI as “*the medium through which the communication between users and computers takes place*” ([Hix et al., 1993](#)). Then, two different definitions of 3DUIs are described by Bowman et al. “*a UI that involves 3D interaction*” ([Bowman et al., 2004](#)) and Molina-Masso et al. “*a user interface in which the language used by the user to transmit information to the computer or the language used by the computer to transmit information to the user, are based on the space and its 3 dimensions*” ([Molina-Masso et al., 2008](#)).

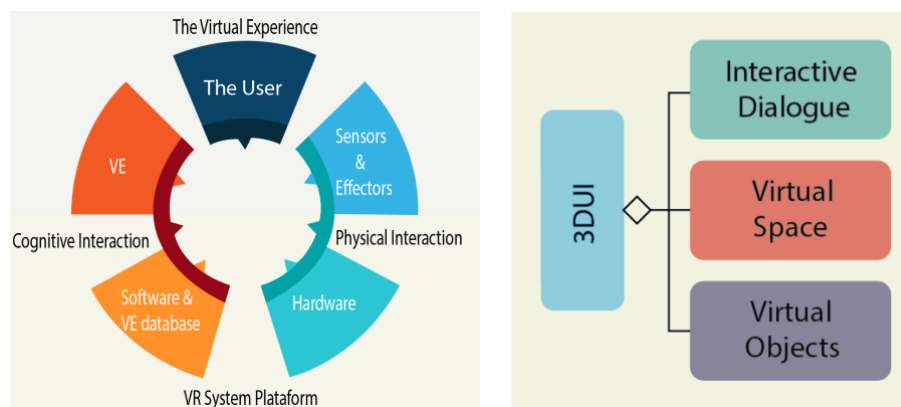
These definitions of 3DUIs are so wide, that they can encompass applications ranging from virtual worlds displayed by 3D graphics (e.g., Head Mounted Displays - HMD), to simply output (e.g., haptic or audio feedback) and input (e.g., body motion, gaze, etc.) modalities involving any kind of 3D interaction. Such a broad definition of 3DUIs results



in a very heterogeneous mix of interfaces, which can involve a great variety of devices (e.g., HMDs, 3D or tactile displays for output; tracking systems or data-gloves for input) and techniques. When compared with 2D-based interfaces relying on well-established devices (i.e., mouse, keyboard and monitor) and metaphors (i.e., WIMP metaphor), 3DUI development is significantly less mature, not only due to the heterogeneity of the hardware involved, but also due to the lack of standardized interaction techniques and methods. This can be illustrated by analysing the current state of 2DUIs.

The WIMP metaphor used by 2DUIs, has been widely accepted, and a standardized set of Interaction Techniques (ITes), Basic Interactive Tasks (BITas) and controls have been identified which are natively supported by all modern operating systems (OS). Even if the first HMD ([Sutherland, 1968](#)) was created approximately at the same time that the first mouse ([Gelbart et al., 1968](#)) (both created in the 60s), the field of 3DUIs still does not rely on a standardized set of interface elements (ITes, BITas, controls) ([Bowman et al., 2001](#)).

In general, the lack of agreement and standardization seems to be a well-known issue in the field and therefore many different classifications have been proposed for 3DUIs. For instance, Marsh et al. classifies the 3DUI elements depending of 3 factors (element's components): (1) imaginary, (2) behaviour and (3) interaction ([Marsh et al., 1998](#)), while Eastgate uses the relationship between the user and the VR system in terms of physical and cognitive interaction levels ([Eastgate, 2001](#)) (see **Figure 2.1** left). However, none of them has been standardized.



**Figure 2.1 Diagrams of 3DUI components: (Left) Eastgate and (Right) Molina.**

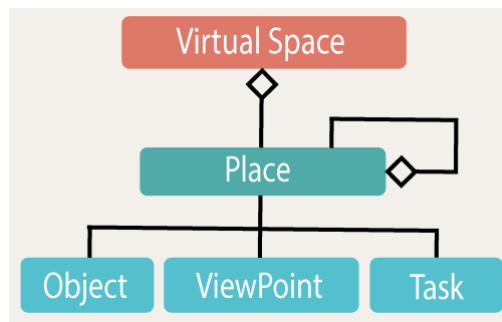
While the number of classifications proposed over the years has been large, we will use the classification by ([Molina-Masso et al., 2008](#)) in this thesis (see **Figure 2.1** right), as this classification was created with the intention to unify such diverging views from

previous work. We then discuss the three main categories in this classification as well as their features and functionalities for 3DUI i.e., *virtual space*, *virtual objects* and *interactive dialogue*.

### 2.1.1 Virtual Space

The *virtual space* is one of the most important and representative elements of a 3DUI. Theoretically, it is an infinite synthetic 3D space that users can navigate and explore, but in reality, it also includes boundaries to limit users to the populated areas and avoiding navigations over empty parts of the space (Bowman et al., 1999).

“RealPlaces” design guidelines (David, 1998) structure a *virtual space* in places and their associated set of tasks. Similarly, IDEAS3D methodology (Molina et al., 2003) proposes a parallelism between GUI elements and VE elements, i.e., different places in the VE are associated to correspondent windows in the GUI and to a set of features to achieve the associated task (e.g., the appropriated point of view to perform the task). Thus, the *virtual space* results in a structure made of aggregated places that form a hierarchical list of tasks to be done in the VE. **Figure 2.2** depicts such structure and indicates the relationship between places, objects, and task (Molina-Masso et al., 2008).



**Figure 2.2** *Virtual space components diagram according to (Molina-Masso et al., 2008).*

Analogous to how GUIs have controls arranged in a 2D canvas, 3D spaces can be considered as the “canvas” (i.e., interface space) of VEs and their 3D elements (i.e., *virtual objects*) as the controls deployed to interact with the system logic. Then, the system logic translate the information from the interface elements (2D controls / 3D elements) to achieve a specific task (e.g., selection, manipulation, etc). This particular description of the space as a user-to-logic interface can be used to describe the *interactive dialogue* between a single user and the system.

However, as proposed by (Martinez-Plasencia, 2010), an extended perspective is needed to describe the *interactive dialogue* in a multi-agent-to-system case e.g., collaborative



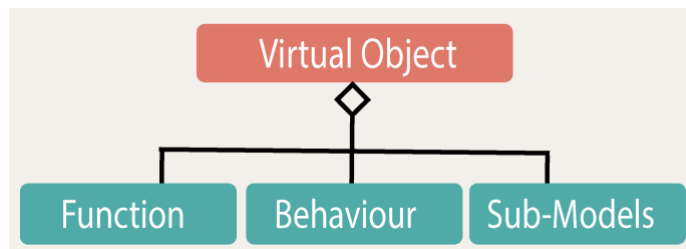
VEs with multiple intelligent agents (e.g., enemies within games). In this case, a better approximation of the concept of *virtual space* would be that considering the space as a medium, enabling entity-to-entity communication. Each entity (i.e., user or agent) is assumed to be autonomous, using the space around it to deploy the controls that allow it to send/receive information from other entities and enclosing its own set of tasks and behaviours.

### 2.1.2 Virtual Objects

Another important component of a 3DUI is the *virtual object*. As mentioned in the previous section, *virtual objects* act as the interface's components of VEs allowing to sustain the *interactive dialogue* between the user and the system. That is, a *virtual space* with no objects will be useless, as no *interactive dialogue* (e.g., tasks) will be supported.

Many authors have proposed classifications of *virtual objects*, based on multiple factors. For instance, based on their behaviour (whether they have it or not) (Sutcliffe, 2003) or based on their structure (passive, active or agents) (Tanriverdi et al., 2001). Others divide them into application objects and interactive objects (Conner et al., 1992). Again, we particularly use the model proposed by (Molina-Masso et al., 2008) as it gathers conceptual information from the afforded mentioned classifications and because of its simplicity and semantic richness.

As shown in **Figure 2.3**, the elements of a *virtual object* according to (Molina-Masso et al., 2008), are composed by: (1) the function which refers to a set of operations, procedures or services that the object offers to other entities in the VE; (2) the behaviour, that controls the execution of the functions according to the object's internal state, the current state of the VE (e.g., time, flags) or other events from the VE (e.g., collisions); and (3) Sub-models that describe objects' features such as visual, auditive or haptic representations, but also position of their interactive parts and physical features (e.g., weight, stiffness, etc.).



**Figure 2.3** Virtual Object components, according to Molina (Molina-Masso et al., 2008).

**Table 2.1** shows the classification of the possible *virtual object* types, according to the elements just described. The resulting *virtual object* types from this classification are:

- **Process:** This object type represents algorithms or functions that have no representation in the user interface. This object type can be modelled as objects without “sub-models” component.
- **Static objects:** They are objects with a representation in the user interface but with no associated “behaviour” component. They are usually used to add context to the virtual scene (to create a specific atmosphere) but they can also be used to provide information cues to users (e.g., a static map can help user's find their way in VR *navigation* tasks).
- **Dynamic objects:** This object type has a representation in the user interface and has an attached "behaviour" component which is autonomous. However, dynamic objects are not interactive elements as they do not react to external events (e.g., a virtual clock behaviour).
- **Reactive objects:** Objects of this type have a representation in the user interfaces and a "behaviour" component too. However, its behaviour is only triggered by external events from the user or from other objects in the VE (e.g., the behaviour of a lamp, turning on/off based on a button press).
- **Agents:** These objects have a representation in the user interface and a specific behaviour component that combines reactive and autonomous behaviours. This is commonly found on video games through AI (artificial intelligence) mechanisms.
- **Avatars:** Particularly, these objects can be considered as a user's container, as the user embodies the object to interact with other objects within the VE (e.g., avatars can be human-like models, robots, cars, planes, etc.).

Object type	Sub-model	Behaviour	Example
Process	No	Algorithms	Gravity behaviour
Static	Yes	No	Tree, scenario
Dynamic	Yes	Not affected by other objects	Clock
Reactive	Yes	Reacts to external events	Button
Agent	Yes	Autonomous behaviour	Enemies (games)
Avatar	Yes	User-driven behaviour	Humanoid, airplane

**Table 2.1** Virtual object types and corresponding features (Molina-Masso et al., 2008).

### 2.1.3 Interactive Dialogue

The previous elements provide the infrastructure (i.e., *virtual space*) and the elements (i.e., *virtual objects*) that support interaction within the VE. These infrastructure elements allow users and objects to interact with the system logic or with each other. However, the internal logic of the system (what interactions are supported and what they do) still needs to be defined. Foley et al., by introducing the “*interactive dialogue*” states that there is communication between the user and the environment that allows them to access the functionality of the virtual world (Foley et al., 1990). Similarly, Molina defined it as “*the communication between the user and the system*” (Molina-Masso et al., 2008). However, as these definitions only consider a single user and the system, a more extended definition was proposed by Martinez-Plasencia to support multi-user applications such as the case of Collaborative Virtual Environments (CVE), i.e., “*the sequences of actions that the entities of the CVE will perform and the reactions that will be executed in response*”. (Martinez-Plasencia, 2010).

Element of 3DUI	Element of natural language
Interactive Dialogue	Sentence
Interactive task	Word (meaning)
Interaction technique	Word (syntax)
Hardware Input	letter

**Table 2.2** Foley’s analogy between interactive dialogue & the elements of a sentence (Foley et al., 1990).

The *interactive dialogue* in 3DUIs can be seen from different levels of abstraction. According to Foley and his comparison between the elements of a 3DUI and the elements of a sentence (elements of the natural language) (Foley et al., 1990), the *interactive dialogue* can be considered as the sentence, composed by: (1) hardware inputs (letters), (2) interaction techniques (word syntax) and (3) interactive tasks (word meaning). **Table 2.2** depicts Foley’s metaphor of *interactive dialogue* using three equivalent components: (1) Input devices; Interaction Techniques (ITes); and (3) Interactive tasks (ITas). In this analogy, input devices (user’s actions) are the lower level of abstractions and they represent the letter to generate a word structure (ITes). The main difference between ITes (word syntax) and ITas (word meaning) is that, while an ITE provides the necessary

information to execute a task (where such task is not known or does not exist), an ITa takes the information provided by the ITe, and computes the appropriate reaction according to a broader context (e.g., the state of the system logic). In the next sections, we explore the *interactive dialogue* based on each level of abstraction (ITes, ITas and input devices).

#### 2.1.3.1 The Interactive Tasks (ITas)

Molina defines an ITa as “*the activity required to achieve a goal and that requires one or more information units as an input*” (Molina-Masso et al., 2008). ITas classify the information units (generated from ITes) and supply them with meaning. Foley classifies ITas into (1) Basic Interactive Tasks - BITas (that can process only one information unit) and (2) Composite Interactive Tasks - CITas (that can process multiple information units from the same or different type) (Foley et al., 1990). This classification allows decomposing complex tasks into simple steps, that Molina called “*operations*” on objects (Molina-Masso et al., 2008). This “*operations*” can be also explained as the combinations of object’s functions and behaviours (see **Figure 2.3**). In other words, the operations are related to the executions of an object’s function.

The “*operation*” concept is very relevant from this level of abstractions as it can contribute to defining the *interactive dialogue* as “an exhaustive definition of the sequences of *operations* required to accomplish the tasks of the CVE” (Martinez-Plasencia, 2010), with *operations* encapsulating: i) the conditions to be satisfied to trigger them; ii) the information to be transmitted (between objects); and iii) the reactions (object functions) to be executed as a result. This *interactive dialogue* description is independent of lower levels of abstractions (ITes and input devices) and better connects with the object structure presented in the previous section. This allows us to generate a description of *virtual objects* (based on their behaviour and interactions) independently of their nature (avatars, agents, etc.).

#### 2.1.3.2 BITas Classification

Foley’s ITas classification (BITas and CITas) was well accepted, while indeed it exists a set of BITas proposed back in 1994 (Herndon et al., 1994), a final classification has not been agreed yet, as many other classifications have been proposed along the years (Foley et al., 1990; Herndon et al., 1994; Bowman et al., 1999; Boyd et al., 1999; Barrilleaux, 2001; Sutcliffe, 2003). To narrow the scope about these classifications and according to the line of this thesis (mainly focused on VR), we discuss only those considering 3DUIs

on the frame of VR applications. **Table 2.3** shows the three main classifications proposed within this scope.

Bowman et al. proposed a classification based on categories (travelling, selection, manipulation and release) and provides a set of BITas to achieve each of them (**Table 2.3** left), (Bowman et al., 1999). Boyd proposed a quite similar classification but considering “release” task as part of object *manipulation* and proposing instead a “data input” task to include voice and keyboard inputs to the system (**Table 2.3** middle). Sutcliffe similarly includes travelling (as movement and *navigation*), selection and manipulation (as *object manipulation* and interaction), but also considers additional tasks such as conversations with agents and non-VR features (**Table 2.3** right).

### 2.1.3.3 Interaction Techniques (ITes)

This is a particularly controversial section, as we defined before, ITes can be the medium to accomplish a certain task, but there is a diffuse line between where an ITe ends and an ITa begins (i.e., some authors consider certain elements as a part of ITes while others do not). According to Foley, an ITe can be described as “*the way input devices are used to generate information*”. Molina considers the avatar as a part of an ITe (Molina-Masso et al., 2008). Additionally, Herndon et al. describe ITes as “*the interfaces that the users interact with in order to accomplish a task*” (Herndon et al., 1994), this definition also includes interactive objects as part of ITes (avatars, 3D widgets, etc.). However, other authors would consider the interaction between these elements as a ITas component (Bowman et al., 1999).

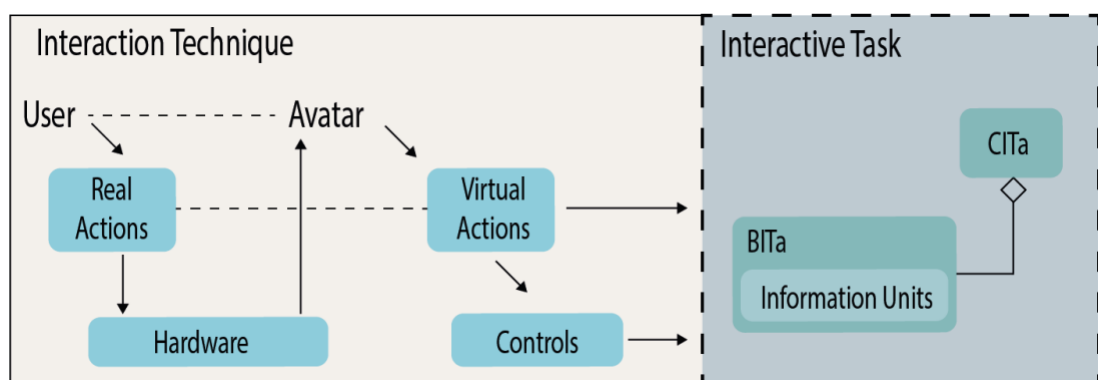
Bowman (1999)	Boyd (1999)	Sutcliffe (2003)
<b>Traveling</b> <ul style="list-style-type: none"> <li>• Select destination</li> <li>• Select velocity</li> <li>• Define conditions</li> </ul>	<b>Navigation</b>	<b>Movement and navigation</b>
<b>Selection</b> <ul style="list-style-type: none"> <li>• Indication of object</li> <li>• Feedback</li> <li>• Indication to select</li> </ul>	<b>Selection</b>	<b>Object manipulation and interaction</b>
<b>Manipulation</b> <ul style="list-style-type: none"> <li>• Object attachment</li> <li>• Object position</li> <li>• Object orientation</li> <li>• Feedback</li> </ul>	<b>Manipulation</b>	<b>Conversation with agents</b>
<b>Release</b> <ul style="list-style-type: none"> <li>• Indication to drop</li> <li>• Object final location</li> </ul>	<b>Data Input</b>	<b>Non-VR features</b> <ul style="list-style-type: none"> <li>• Data input</li> <li>• GUI-like interaction</li> </ul>

**Table 2.3 Basic Interactive Tasks (BITas) and data types classification by (Bowman et al., 1999), (Boyd et al., 1999) and (Sutcliffe, 2003).**

If we consider (i) the disagreement about the ITes' boundaries, (i) the massive diversification of input devices and (iii) the complexity of 3DUIs when compared with traditional GUIs (Herndon et al., 1994), all together may explain why a standardized set of ITes has not been identified yet.

Acknowledging this problem, Bowman et al., proposed a correspondent set of generic ITes to be used to accomplish each ITa (Bowman et al., 1999). However, because of the increased number of ITes available, choosing the appropriate ITE to achieve a specific ITa is not trivial. For instance, LoUISE laboratory has conducted several testbed evaluations (Bowman et al., 2002) to help designers with ITes identification according to system features (García et al., 2005; González et al., 2009). Another classification focused on manipulation techniques, was done by Poupyrev et al., highlighting some ITes metaphors such as virtual pointers, virtual hand, etc. (Poupyrev et al., 1998). These metaphors are very relevant as they represent the mental model that users build when interacting with a specific ITE. This generated mental model help users to understand what is allowed and constrained by the ITE.

Previous attempts to classify and generate a standard set of ITes that 3DUIs can use, have been done to enable the implementation of generic toolkits for ITes (as those established for GUIs). Because of the complexity to achieve the goal, a generic model to describe the process of an ITE to assess 3DUIs developers becomes crucial. Foley suggested to study the degrees of freedom (DoF) from input devices and associate them to the ones each ITa may require. Elsewhere, (Molina-Masso et al., 2008) proposed a more complex model that considers the user's action in the real world, going through input devices, until the action is performed by the virtual avatar (see **Figure 2.4**).



**Figure 2.4** Model proposed by Molina to describe the ITes' stages, and their boundaries with ITas.

This model clearly defines the boundaries between ITes and ITas. For instance, it defines ITes domain from the process of users' real actions to the avatar's virtual actions (not including the interaction between *virtual objects*). The "controls" in the model (see **Figure 2.4**) are modelled as reactive objects (those are an abstraction of widgets proposed by (Herndon et al., 1994)), that translate user's actions to information that ITas will require. Virtual actions are then, the connection between ITes and ITas, in the way that, ITes generate virtual actions that the system analyses to select the BITa to be executed (giving a meaning to the ITe).

#### 2.1.3.4 Physical Devices

From the theoretical point of view, *physical devices* are the tools used to measure user actions (e.g., users' gestures, body position/orientation/velocity, typed text, joystick movement, etc.) and translate them to system inputs. These devices are an important component of the implementation of an ITe. As Bowman et al. states, different devices can be used to implement the same ITe, and the same device can be used to implement different ITes (Bowman et al., 2001). Sutcliff classifies *physical devices* in *discrete* and *analog*, while Bowman uses *discrete* and *continuous*. However, Martinez-Plasencia highlights the ambiguity of the term *continuous* depending on the applications (the device's sampling resolution) and proposed to use devices' updates per second instead as a more objective parameter (Martinez-Plasencia, 2010).

Furthermore, Foley proposed to classify *physical devices* based on the degrees of freedom they can generate (DoF). While Molina proposed the concept of "abstract input device" that considers the number of DoF and input type (discrete or continuous) to generalize the suitable devices (those that meet the requirements) to implement a given ITe. Additionally, Martinez-Plasencia considers including the device input modality (e.g., voice. Motion, BCI, etc.) to complement Molina's concept. The same concept can be also applied to output devices (abstract output devices) to describe the modality to be used as feedback to the user (visual, audio, tactile, etc.) and the elements to provide such feedback.

However, from the practical point of view, input devices have had huge advances in general quality. For instance, overcoming along the years well-known limitations (e.g., render quality, perception, low frame rates, latency, etc.), increasing also their acceptability by making their acquisition into affordable prices (allowing to spread their use to public). However, this increment in the generation of better and more affordable



input devices is, in turn, increasing the heterogeneity of the hardware, making their standardization even more complex. Although attempts to standardize hardware such as HMDs and tracking systems (e.g., from frameworks such as Unity, OpenVR among others), they are far to become a standard. While great advances in hardware input have been done, these advances are focussing on the improvement of generic issues (low frames per second, rendering, latency, etc.). While they help on providing the necessary hardware power to ITes when needed to achieve a task, being improvements at hardware level (potentially for any 3DUI), they cannot exploit the specifics of the task supported by a concrete application to further increase the richness of interaction. This highlights the need for better interaction techniques that may take advantage of these hardware improvements, but also take advantage of the knowledge of the tasks that the user will perform to enable more natural and intuitive interactions with systems.

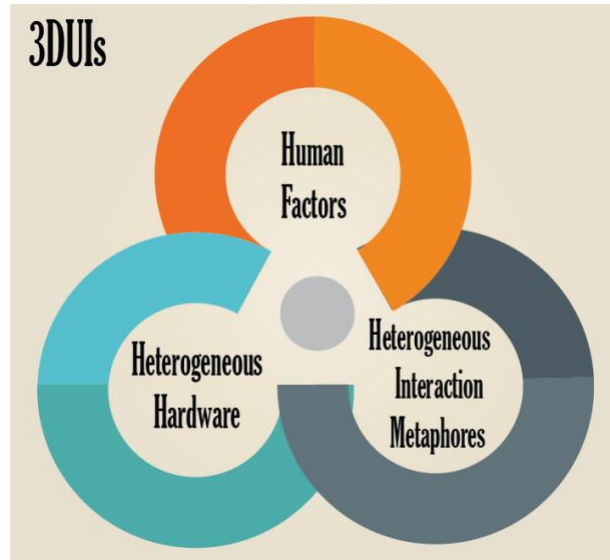
## 2.2 Opportunities and Justification of the Approach

Even when traditional 2DUIs have a well-defined set of ITes, ITas, controls and devices, such as the WIMP metaphor ([Van Dam, 1997](#)), they fail in exploiting the human potential. As depicted (**Figure 1.1**) i.e., the “way the system sees us” can be very limited ([O'Sullivan et al., 2004](#)). In contrast, 3DUIs have a much higher potential to exploit human capabilities by stimulating all the senses ([LaViola Jr et al., 2017](#)), but also, taking advantage of the way we explore and interact in the real world enabling an easy transition from real to virtual interactions (decreasing the learning process) ([Bricken, 1991](#)), allowing also the generation of more natural and realistic interaction paradigms.

Despite the 3DUIs potential, the input requirements (to make extensive use of humans’ perceptual and interactive potential) lead to a high heterogeneity of devices and interaction techniques that increase the interaction complexity and turn, the standardization of 3DUI elements is even more challenging, reflecting thus the non-mature state of the field ([Bowman et al., 2001](#)).

While the improvements on hardware have been remarkable in the last years, the state of current software/ITes has not improved that much as illustrated in this chapter. Relevant conceptual background about this topic exists, however, most of the representative ITes and ITas were proposed long time ago (e.g., virtual hand and ray-casting) ([Bowman et al., 1999](#)). Thus, these approaches may not take advantage of the computational capabilities from current hardware. Also, as those were designed as general metaphors, they do not exploit the interplay between *human factors* and specific ITas.





**Figure 2.5** Diagram to illustrate our aim to better exploit *human factors* in the generation of more natural 3DUIs, by considering the current state of hardware and Interaction metaphors.

**Figure 2.5** shows a representation of such interplay of these factors in 3DUIs, based on the current state of heterogeneity of hardware and interaction metaphors, and the need of greater inclusion of *human factors* to better benefit from the human potential for interaction. On the basis of this model, our goal is to better exploit *human factors* to create more natural interfaces that better exploit human capabilities. Current computers provide a vast amount of computational power which is not fully exploited by generic ITes and ITas. This power would allow the use of more advanced algorithms (e.g., computational approaches, real-time simulation) to produce ITes and ITas that provide better support for 3DUI interactions. Seeking this opportunity, we combine *human factors*, mathematical formalizations and computational methods (i.e., optimization methods and real-time/offline simulations).

Since, it is impossible to fully explore the 3UIs scope, we explore the potential of this approach by reviewing each of the main stages of the interaction. **Figure 2.4** shows the boundaries between ITes and ITas as proposed by Molina ([Molina-Masso et al., 2008](#)), where we clearly observe the interaction's stages involved (mappings, ITes and ITas): First, the generation of mappings to enable the *dialogue* between input devices (user's actions) and the system. Second, how the generated information is converted to avatar's actions and controls and; Third, this resulting data is transformed into *information units* that a corresponding BITa will use. In this thesis, we used Molina's model (**Figure 2.4**) to illustrate our exploration through the interaction stages.

### 2.2.1 Mapping Generation

In the [Chapter 3](#), we explore the integration of *human factors* in the generation of input mappings for a complex task (e.g., text entry), by first, mathematically formalizing expert's tacit knowledge (*high-level* factors) and quantifiable factors (*low-level* factors) and second, combining these formalizations with offline simulation and optimization methods to generate gestural mapping that better reflects human capabilities and limitations.

### 2.2.2 Interaction Techniques (ITes)

On one hand, [Chapter 4](#) explores how to improve *human factors* such as ergonomics in VR object *manipulation* tasks. We combined the most natural object manipulation metaphor in VR (virtual hand) with unnoticeable distortions in users' perception about interactable 3D elements positions, generating thus, an interaction technique that combines comfort mappings (from a real-time ergonomic analysis), a movement redirection method and an optimization strategy to enable more comfortable object interaction.

On the other hand, [Chapter 5](#) explores how to maintain a natural navigation metaphor while avoiding space limitations (due to either, a limited real space or a constrained tracking system). We mathematically modelled users' displacement within the VE, to dynamically scale it based on an *allocentric* navigation technique to generate the illusion of exploring bigger spaces. Our technique maintains a natural interaction in relevant zones (areas with 1:1 movement ratio for precise interaction and fine manipulation or manoeuvring tasks), while at the same time compresses the non-relevant spaces generating navigation fields.

Even with the observed benefits from this technique, we observed a *Drift* effect that interestingly was present in other *adaptive-scale* navigation techniques. This *Drift* effect tends to grow over time decoupling the centres of the virtual and real worlds i.e., shifting the virtual targets to a possibly non-reachable position in the VE. Therefore, [Chapter 6](#) explores this *Drift* effect, by mathematically modelling its behaviour. We then used real-time simulations to generate a correction method that significantly reduces the *Drift* effect without disrupting the navigation technique scaling policy. This correction method leverages the use of more natural navigation techniques (using physical locomotion) while addressing the problem of limited real space exploiting thus human capabilities of real walking.

### 2.2.3 Interactive Tasks (ITas)

[Chapter 7](#) explores *object selection*, which has been identified as a fundamental task in 3DUIs ([Mine, 1995](#); [LaViola Jr et al., 2017](#)). Here, we introduced a method that mimics the way users interact with physical surfaces in the real world. We combine 3D *selection* in mid-air and 2D *selection* on a touchscreen tablet in VR. Using this hybrid approach enables more comfortable interaction (addressing fatigue common in mid-air interaction) while producing more precise selection (on a constrained physical surface) in dense VEs (e.g., point cloud). To achieve this, we take advantage of the current computational power to enable real-time interaction in big data sets (~6M objects) by a GPU-based approach. This approach aims to enable natural interaction (such as a *pen-and-tablet* metaphor) to provide extra visualization from the VE on a tablet viewpoint, enhancing scene understanding in dense environments, and providing haptic feedback and stability in VR interactions.

### 2.2.4 About Isomorphic and Non-isomorphic Mapping

All the approaches proposed in this thesis were aimed to enable the generation of more natural 3DUIs that take advantage of human capabilities, but also considering human constraints (e.g., ergonomic limitation) to generate more human-tailored 3DUIs. Although not intentional, our exploration soon led us to explore the mapping between real and virtual spaces, and to the identification of the advantages in the use of isomorphic and non-isomorphic mappings, depending of the specific tasks and the *human factors* involved. The isomorphic view relies on a strict one-to-one (1:1) analogy between body movements in both real and virtual worlds, while the non-isomorphic view allows real-virtual movements discrepancies at the expense of realism ([Poupyrev et al., 1996](#); [Pierce et al., 1999](#)). Although 1:1 analogy has been suggested to be more natural ([Knight, 1987](#)), such mappings are often impractical due to input devices constraints (e.g., when tracking ranges are restricted). Therefore, non-isomorphic techniques provide a solution, although deviating from realism, they maintain usability and performance ([Bowman et al., 1997](#); [Poupyrev et al., 1997](#)).

[Chapter 4](#) explores solutions to address user fatigue in mid-air manipulations by modulating the isomorphism of interaction (through a movement redirection technique). We take advantage of isomorphic mappings to improve comfort while maintaining usability and feeling of control. The implicit benefit of isomorphism in this specific application is that the objects remain fixed at specific positions from the user, even if

these positions do not match with visual positions, this still allows users the use of muscle memory to reach them (as the body remembers where the object was, from previous interactions) highlighting its potential for object *manipulation*.

Moreover, we also employed non-isomorphism to address space constraints in virtual *navigation* (in [Chapter 5](#) and [Chapter 6](#)) i.e., the non-isomorphic mapping was the key to allow navigation along larger areas, whereas [Chapter 6](#), shows how this non-isomorphism needs to be controlled to avoid users drifting out of navigable spaces.

In this chapter, we have reviewed some important concepts, components, and classifications of 3DUIs. This review aimed to ground the basis where our general approach is built on. However, specific related work is provided at the beginning of each chapter to better reflect the scope of both, the identified issue in the correspondent interaction stage (see **Figure 2.4**) and the way we address it.

## Chapter 3

# Designer Led Computational Approach to Generate Mappings for Devices with Low Gestural Resolution

According to our thesis structure, in this chapter we explore hardware *input*, i.e., the first step of the interaction stages (**Figure 2.4**), and the first element of the *interactive dialogue* (Foley et al., 1990). We started with this stage, since data *input* is considered the link between virtual and real worlds (LaViola Jr et al., 2017). More effective communication between devices and the system (*input* mappings) can strongly impact the naturalness of the interaction, user performance and overall user experience. Next, we introduce a hybrid technique that addresses mapping complexity of gestures *input*, leveraging the combination of *human factors* and computational approaches i.e., exploiting the close coupling between *human factors* and ITes.

### 3.1 Introduction

Gestures play an inherent role in our everyday communication, to the extent that we make use of them even when our interlocutor is not present, such as when speaking on the phone (Rimé, 1982). Gestures can be used to communicate meaningful information (*semiotic*), manipulate the physical world (*ergotic*) or even to learn through tactile exploration (*epistemic*) (Cadoz, 1994). *Semiotic* gestures have been of particular interest to the HCI community as a powerful way to communicate with computers (Rimé et al., 1991; McNeill, 1992).

The design of interfaces involving gestural interaction remains a challenge. First, advances in hardware have been remarkable. Gestural interaction is no longer restricted to data-gloves (Kessler et al., 1995; Fabiani et al., 1996; Weissmann et al., 1999), and there is an increasing range of potential devices, allowing gesture tracking on un-

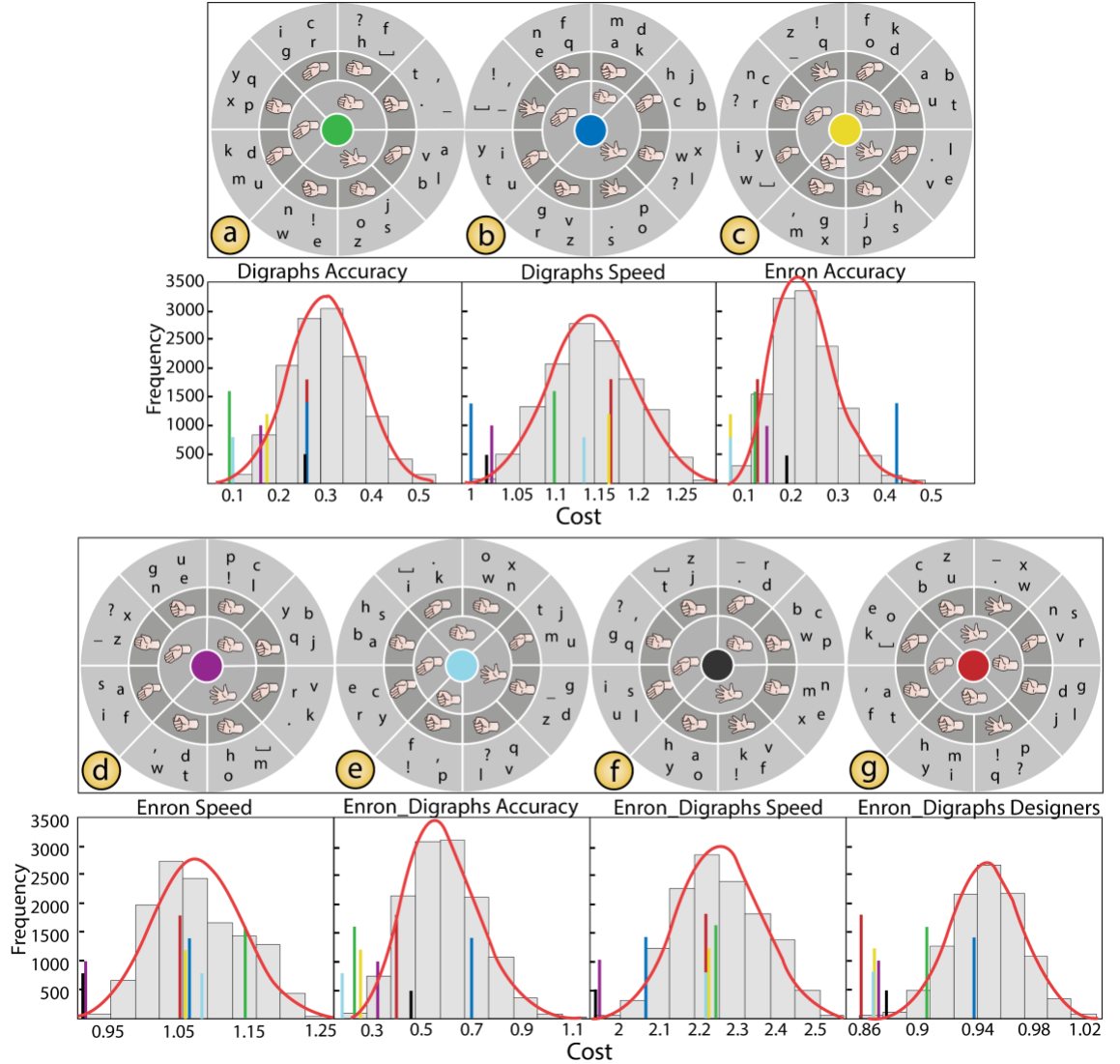
instrumented hands or even in mobile formats. Secondly, the methods to design these experiences have followed a much slower progression, not coping with the increasing number of devices available, and still relying on iterative methods and designers' expertise (Sturman et al., 1993; Gabbard et al., 1999). In other words, in the last years the community has been focused mainly on generating new devices to improve hardware capabilities that could be reflected in better interaction with the systems (e.g., release of new input devices, tracking techniques, etc.). However, there has been a lack of effort in advancing the understanding on designing experiences involving gesture interaction, for instance considering *human factors*, hand and gesture tracking allowances, comfort, mapping to tasks, etc.

As a result, interaction designers are faced with a very challenging task, with many factors involved in the creation of the gestural interface. While some factors will be easy to assess (i.e., device's comfort, accuracy, speed), others will be more complex (i.e., social acceptability and cognitive load). Particularly challenging is the elicitation of the most appropriate gestures and their mapping to tasks, which can easily lead to a combinatorial explosion. For instance, our example case study (text entry) offers more than 35K ways to map gestures to input commands and more than 12K ways to map these to actual letters. While iterative methodologies, designers' intuition and heuristics might help, it will be costly to navigate this vast solution space and identify the optimum *interactive dialogue*. In contrast, computational approaches might struggle to capture the complexity of the task, particularly for complex/subjective factors.

Unlike previous methods, we propose a hybrid approach, merging designer-led methods and computational approaches for the generation of robust gestural mappings under such challenging conditions (i.e., large solution space involving complex high-level factors). More specifically, we present an expert-guided, semi-automated design of *interactive dialogues* for low gestural resolution devices. Our approach consists of four steps: **i)** *quantify low-level factors* (error, speed or accuracy); **ii)** *semi-structured workshops with designers* (identify higher level factors, such as cognitive load and experts' heuristics); **iii)** *formalization & optimization* (using objective and designers' knowledge to produce a mathematical model, and computing optimum mappings); and **iv)** *comparative evaluations* (to guide the iterative interface design, in a cost-effective manner).

We demonstrate this approach applying it to the design of a text entry technique using a Myo device. **Figure 3.1** (g) shows the result – a multi-level mapping between the input

gestures and characters for text entry. To assess the value of our approach, we compared the mapping produced from our hybrid approach (incorporating designers' *high-level* factors) to several purely computational, naïve mappings. Particularly, we defined 6 alternative cost functions (i.e., models to assess the quality of a mapping) optimizing for time and accuracy, and explored up to 2.7 billion possible mappings, finding the optimum mapping for each of the 6 naïve cost functions.



**Figure 3.1** Resulting mappings from the full optimization using different training database and cost function's factors. Below each layout, its histogram is shown. The cost per layout is represented along all histograms using colour code (M\_C1=green, M\_C2=blue, M\_C3=yellow, M\_C4=magenta, M\_C5=cyan, M\_C6=black and M\_D=red).

**Figure 3.1** shows histograms for all these mappings according to: the naïve computational metrics (a-f) and our approach (g). The optimum mappings computed are also highlighted within each histogram (bars). These show that, while naïve functions are highly ranked according to the designers-led metric (i.e., low scores, in **Figure 3.1(g)**),



the designers-led mapping ranked relatively poorly according to each of the 6 naïve cost functions used (red bar showing high values in **Figure 3.1(a-f)**). This could either point towards designers’ insight being irrelevant (or even harmful) or to computational methods failing to capture the complexity of the task. The results from our study show that the designers-led mapping showed a good balance on performance in all factors involved (speed, accuracy, comfort, memorability, etc.) and consistently performing better than purely computational mappings. This reveals an untapped power in the designers’ ability to identify a good cost function, with our approach helping to produce a suitable formalization to exploit the exploratory potential of computational approaches.

We finish this chapter reflecting on these results and on how they should open a discussion on the added value of designers’ intuition and heuristics when exploring gestural interfaces, and the need to make these an integral part of current design methodologies, for large solution spaces.

## 3.2 Related Work

### 3.2.1 Gestural Input Devices: A growing landscape

An increasing number of device options are available to support gestural interaction. Early instances included data gloves and tracking systems, mostly used for Virtual Reality ([Weissmann et al., 1999](#)) and multimodal interaction. These provide high gestural resolution (i.e., high number of distinct gestures), but require user instrumentation, hindering their applicability (i.e., users cannot simply walk-up and use them, wires limit mobility, etc). Wireless tracking systems (e.g., Kinect) can improve applicability ([Dutta, 2012](#)), but their sensors are typically fixed, constraining the user to specific working spaces.

Mobile solutions have also been proposed. Kim et al. ([Kim et al., 2012](#)), presented a wrist-mounted optical system, allowing for hand gestural interaction. Myo armbands use Electromyography (EMG) to record and analyse electrical activity, allowing lightweight mobile gestural input, without hindering the use of our hands and avoiding self-occlusion problems. EMPress ([McIntosh et al., 2016](#)), combines EMG and pressure sensors, providing the same affordances of Myo bands, but with improved gestural resolution. Solutions to extend smartwatch interaction with around device gestural interaction have also been explored ([Knibbe et al., 2014](#)), but they either provide limited gestural resolution ([Kerber et al., 2015](#)) or involve instrumenting the user’s gesturing hand ([Zhang et al., 2016](#)).



### 3.2.2 Gestures and Mappings: Point Studies

The HCI literature has produced a plethora of studies, which can help designers deal with the increasing number of device options available. Sturman et al. ([Sturman et al., 1989](#)) explored and provided guidelines to improve gestural interaction in VR. Studies from Rekimoto ([Rekimoto, 2002](#)), Wu & Balakrishnan ([Wu et al., 2003](#)) provide insight in the context of interactive surfaces, and Grossman et al. ([Grossman et al., 2004](#)) explored the topic in the context of 3D volumetric displays, just to mention some. However, these illustrate how information related to gestural interaction is scattered across individual point studies, focused on specific tasks and contexts.

A more general approach to designing gestural interaction has been to formalize user elicitations ([Gelain et al., 2010](#); [Kammer et al., 2010](#)). Designers seek end-user input on mapping gestures to tasks, classifying gestures into high level groupings based on salient properties (e.g., direction of movement, finger poses, etc). Elicitation studies have been successfully used in a number of contexts, but have also been criticized for biasing results by basing them on input from populations unfamiliar with the task or capabilities of a device ([Gelain et al., 2010](#); [Davey et al., 2015](#)).

Alternatively, designers can gain insight about the mapping between gestures and tasks from related literature. Focusing on text entry (closest to our case study), the QWERTY keyboard serves as a preeminent example of discrete mapping, enforcing a 1:1 mapping between each key (gesture) and a letter (task). It also illustrates a mapping designed around the mechanical limitations of past typing machines, rather than its appropriateness for human input.

Computational approaches have proved to be valid tools to identify better mappings. Bi et al. showed clear improvements for clarity (avoid gesture ambiguity) and typing speed for the most common digraphs in English ([Bi et al., 2016](#)) by simply swapping two keys (I and J). Bi et al. ([Bi et al., 2010](#)) explored alternative mappings by swapping a few neighbouring keys, to get a layout with better performance on speed, while retaining QWERTY similarity. Smith et al. showed a similar approach, improving clarity, speed and QWERTY similarity for 2D gesture typing. Alternatives for situations where 1:1 mappings are not available (e.g., mobile phones) have also been tackled using computational approaches, mostly through predictive text entry models ([Pavlovych et al., 2004](#); [Gong et al., 2005](#)). Other works have focused on exploring the extent of human hand's dexterity, creating mappings that benefit from all its bandwidth. Sridhar et al.

explored the bio-mechanical features of the hand (flexion levels, inter-digit dependencies) (Sridhar et al., 2015), while PianoText (Feit et al., 2014) leverages users’ musical skills, using a piano keyboard and chords to create an ultrafast text-entry system. In all cases, the benefits of computational approaches are limited by the use of low-level, quantifiable factors.

Thus, while designers are usually in charge to explore such large solution space, they can introduce user biases. The emerging alternative within HCI is the use of computational approaches, but these might fail to capture higher level aspects of such complex tasks as they tend to bias/limit their results towards quantifiable factors that are easy to assess). Our approach intends to bridge this gap, being the first one to put together the benefits of both approaches (designers-led vs computational solutions), by blending designers’ methods/insight and computational approaches.

### 3.3 Our approach: Semi-automatic Mappings for Low Input Resolution

Our method aims to bridge the differences between designer-led and computational solutions, capturing designers’ tacit knowledge of the domain, and formalizing it to be exploited by computational approaches. We thus combine *quantitative parameterization of relevant factors* with *domain expert knowledge elicitation*, into a structured approach. We refine these into a *formal model* quantifying the quality of each mapping and using a *global optimization algorithm* to explore the solution space, finding (potentially) the best solution. Our approach is compatible with iterative methodologies and can be seen as the tasks required for one iteration cycle. The outline of our approach can be divided into four stages:

#### 3.3.1 Quantification of Low-level Factors and Constraints

This stage involves the experiments and in-lab tests required to measure and quantify *low-level factors* and *constraints*. *Low-level factors* are simple parameters (e.g., time, errors) associated with the device or modality that might influence the design of the mapping and are easily quantifiable. *Low-level constraints* represent limitations within the device or the way it is used. Using our case study as an example, *factors* can include time to perform each Myo gestures, while excluding the double tap gesture due to its low accuracy can be an example of a *constraint*.

These quantified values will be used in the two following stages: First, they will inform designers, to help produce mappings and formulate heuristics; Second, they provide quantifiable data, used by our optimization methods.

### 3.3.2 Domain Expert Knowledge Elicitation

We use small teams of experts as a way to elicit the relevant factors that need consideration to design the *interactive dialogue*. Different methodologies can be used (e.g., workshops, elicitation studies, prototypes), which help addressing a broad spectrum of aspects that cannot be covered by computational approaches alone (e.g., interface design, feedback elements, definition of the *interactive dialogue*, etc.).

However, while designers must consider the mapping of gestures to tasks, the ultimate intent of this process is *not* the specific mapping chosen (computational searches should help making this specific choice). Instead, we focus on the designers' rationale that they use to determine what might be a good choice of gestures and mapping.

We reflect this rationale as *constraints* (i.e., conditions that must be obeyed) and *high-level factors* (i.e., non-obvious aspects or heuristics affecting interaction, such as social acceptance). These will help our following formalization process and the weighting of the relative importance of each of these factors.

### 3.3.3 Formalization & Optimization

In order to optimize our mappings, we first need to provide a metric for the quality of any given mapping. We formalize the quality of a mapping  $\mathbf{M}$  as a cost function  $\mathbf{C}$  computing as a weighted average of the factors identified by experts, with lower values identifying better mapping:

$$\mathbf{C}(\mathbf{M}) = \sum k_i \cdot \mathbf{Factor}_i(\mathbf{M}) \quad (3.1)$$

The different factors are all normalized to a homogeneous range  $[0, 1)$ , according to the maximum and minimum values observed from the quantification. The value for  $k_i$  (influence of a given  $\mathbf{Factor}_i$  in the mapping  $\mathbf{M}$ ) needs to be estimated from the experts' impressions and analysis (further details follow). This assures that the contribution of each factor to the quality of  $\mathbf{M}$  is the result of the designer's insight, and not the result of the factors' relative orders of magnitude. In our example, the sum of factor weights ( $\sum k_i$ ) equals one (factor as a percentage), but any other weight distribution reflecting the expert's impressions can be used.

We then use a global optimization method to explore the solution space, converging towards an optimized solution given the factors and weighing values identified. Although our case study used Simulated Annealing ([Kirkpatrick et al., 1983](#)), other optimization approaches can also be used.

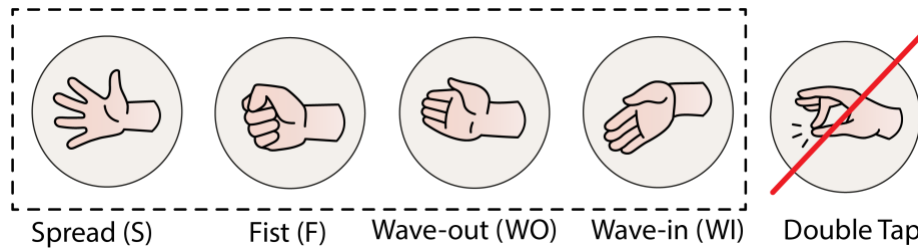
### 3.3.4 Comparative-Summative Evaluation

While the normalization of the factors identified follows quantitative criteria, the estimation of the weight distribution ( $k_i$ ) does not, and relies on the subjective assessment of domain experts. Different weight distributions might reveal different ways of thinking about the solution (e.g., how more relevant is minimizing time over cognitive load?). Computing optimized mappings, according to different weight distributions, and comparing them through comparative evaluations can allow for the best mapping to be identified. This reduces the exploration of the solution space to a few candidates (each resulting from a different weighting strategy), and integrates easily with iterative methodologies for gestural interaction, such as ([Gabbard et al., 1999](#)).

## 3.4 Case Study with Myo: Compute Vs Design

We tested our approach using a Myo device (i.e., very low gestural resolution) for a text-entry task, both as a worst-case scenario and as an obvious match to Foley’s analogy between natural language and a general *interactive dialogue*. The in-built IMU was not used and only the muscle activation was considered. This reduces our gestural resolution even further (more challenging solution space) but it also lends itself to interesting application scenarios. IMU-based gestures are defined relative to the body, and might be restricted during our daily life (i.e., while sitting in a bus, walking or inside a busy elevator). In contrast, our gestures remain relative to the hand, being still available in any situation where the wrist can be moved.

















Finally, we also wanted to assess the added value of our designers’ guided approach when compared to unconstrained computational approaches, based on observable and quantifiable factors alone. We replace the last stage of the method **(iv)**, by a description of the naïve computational mappings used, and a comparison against the results provided by these alternative approaches.



**Figure 3.2 Gestures possible with a Myo armband. We used the enclosed gestures in this work.**

### 3.4.1 Problem Delimitation

Although Myo supports up to five gestures, at the time when this work was carried out “Double tap” was a recent addition with known inconsistencies in its detection ([Thalmic-Labs](#)). Also, any fast and consecutive pair of gestures was detected as “Double tap” (i.e., false positives), conflicting with the use of other potential gesture chains. For that reason, only the four remaining gestures were used (see spread (*S*), fist (*F*), wave-out (*WO*) and wave-in (*WI*), in **Figure 3.2**). We quantified the performance of 16 possible 2-step chain gestures (consecution of two gestures, as in Fig 3). Such 2-step chains require an intermediate relax action (i.e., hand returning to a neutral status between gestures) to be recognized by the system.

Opposite	Orthogonal		Repeat
WI - WO 	WI - S 	WI - F 	WI - WI 
WO - WI 	WO - S 	WO - F 	WO - WO 
S - F 	S - WI 	S - WO 	S - S 
F - S 	F - WI 	F - WO 	F - F 

**Figure 3.3 Two-step chain gestures under designers’ categories.**

We asked our designers to categorize the 2-step chain gestures and they identified three different groups: *opposite*, *orthogonal* and *repeat*. *Opposite* chains combine gestures that activate opposing muscles. *Orthogonal* chains invoke *orthogonal* muscle groups; and *Repetitive* chains contain two instances of the same gesture (see **Figure 3.3**). For example, WI+WI is a *Repeat*, WI+WO is of type *Opposite*, and WI+F is type *Orthogonal*. We will borrow this for the analysis in this section (even if the distinction only appeared during

the later workshops), as its analysis allows us to assess to what extent designers' insight reflects trends in data, or if some aspects pointed by designers would be likely to be included or ignored by alternative purely computational approaches. Finally, we also conducted a similar study for 3-step chain gestures. However, designers soon disregarded these chains during the later workshop (only use 2-chain gestures – **C1**), so our results for 3-step chains are omitted here for brevity.

### 3.4.2 Quantification of Relevant Factors.

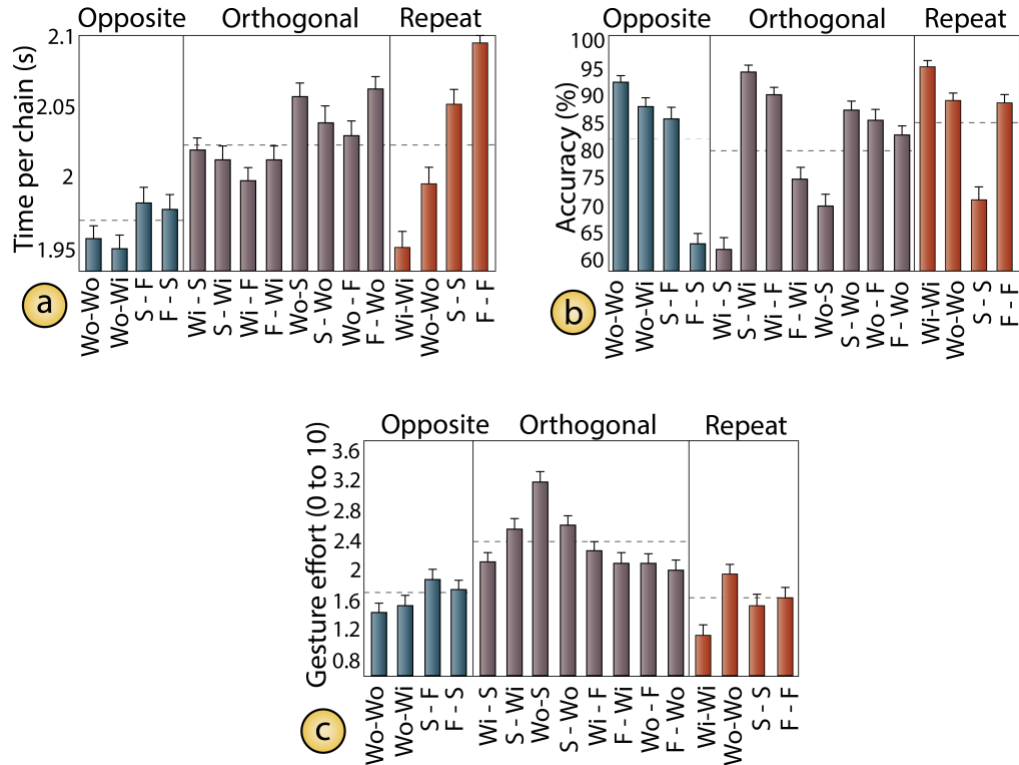
We conducted a quantitative study, where participants performed a series of 2-step chain gestures under different input speeds to evaluate potentially relevant factors (i.e., errors, ergonomics, and preferred 2-step chain gestures). We calibrated the Myo for each individual participant and allowed them to become familiar with the 4 Myo gestures (**Figure 3.2**) and our 2-step chain gestures (**Figure 3.3**). They were then asked to perform the 2-step chain gestures shown on a display, which changed at regular speeds (i.e., each single gesture shown during 0.6s, 0.8s, 1.0s or 1.2s). Participants were asked to complete the gestures accurately and within the length of the prompts, which helped us identify the appropriate “typing speed”.

The experiment consisted of 4 blocks (one block for each input speed) including three repetitions of each of the sixteen 2-step chains gesture, resulting in 192 trials per participant. To avoid participants fatigue given this number of trials, each block was designed to be completed in about 4min giving participants a 3min break between blocks. Due to fatigue could potentially affect participants' performance, we ensured that each block duration was short with enough time to rest. The full experiment duration was then about 30 min, including calibration, training and breaks between blocks.

We counterbalanced the order of the input ratios using a Latin Square design, but gesture order was randomly selected. Time per gesture chain and accuracy (whether the gesture was recognized by Myo or not) were recorded. After each block (i.e., input speed), participants also filled in a Borg CR10 Scale ([Borg, 1990](#)) questionnaire (i.e., specially designed to quantify perceived exertion and fatigue ([Borg, 1990](#); [Robertson et al., 2003](#))) for each of the 16 2-step chain gestures. The experiment was performed by twelve participants (4 females), with average age of 23.53 (21 to 30) SD=2.98, with the study being approved by the local ethics board. Recruitment criteria was: i) all participants right-handed; ii) normal or correct-to-normal vision; iii) no injuries on their hands and wrists; and iv) no prior experience with hand gesture interaction. Outliers were removed

from the data (i.e., mean  $\pm$  2 standard deviation), filtering out 129 trials (5.59 % of samples). We then conducted factorial repeated measures ANOVA ( $p=0.05$  to determined significance) on the factors measured, which we report in the following subsections.

**Time per gesture (F1):** **Figure 3.4(a)** shows the results of time for each 2-step chain. This analysis revealed significant effects of gesture type on time performance ( $p<0.001$ ), justifying its later inclusion as a factor (F1), even for a purely computational approach. Post-hoc tests with Bonferroni corrections show significant differences between certain gestures (e.g., WI+WO vs F+WO,  $p=0.03$ ; WI+WI vs F+F,  $p<0.001$ ), but the high number of pairs to compare (120), made such analysis poorly informative. Therefore, we did analyse time performance based on the categories proposed by the designers (*Repeat*, *Orthogonal* and *Opposite*). *Opposite* gestures performed best (M=1.965s; SD=0.229s), with significant differences ( $p<0.001$ ) between the duration of *Opposite* and *Repeat* gestures (M=2.022s; SD=0.255s) and also between *Opposite* and *Orthogonal* gestures (M=2.028s; SD=0.240s;  $p=0.001$ ). On the other hand, clustering techniques (for time, accuracy or comfort) did not lead to identifying these categories. Thus, this is considered designers' tacit knowledge and would not be considered by purely computational approaches.

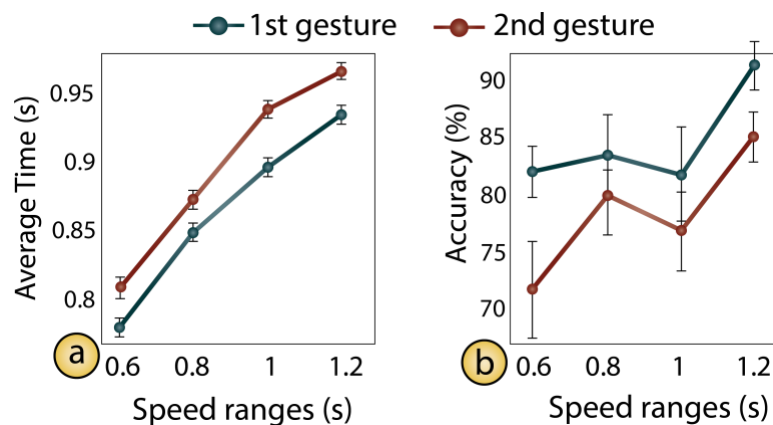


**Figure 3.4 Mappings' performance:** (a)Time per chain gestures for *Opposite*, *Orthogonal* and *Repeat* categories (Mean in seconds); (b) Accuracy per chain gestures (Mean in %); (c) Effort results per chain gesture.



*Accuracy per gesture (F2)*: **Figure 3.4(b)** shows our results for accuracy, revealing overall accuracy is low (70% - 90%). An ANOVA analysis revealed an effect of gesture on accuracy (used as factor **F2**). Again, significant differences were found between specific pairs of gestures, but we focus the analysis on designers' categories. We only found significant differences between *Repeat* ( $M=86.8\%$ ;  $SD=21.57\%$ ) and *Orthogonal* categories ( $M=81.28\%$ ;  $SD=24.45\%$ ;  $p=0.032$ ), but with reduced effect size. Also, no clear patterns could be observed by looking at the categories (values well above and below the mean are present in all categories, in **Figure 3.4(b)**).

*Gesture Comfort (F3)*: Comfort was rated by participants using a Borg CR10 Scale (**Figure 3.4(c)** shows the average of participants' effort per gesture). According to their answers, we found *Repeat* gestures as the most comfortable ( $M=1.5$ ,  $SD=0.33$ ) followed by *opposite* gestures ( $M=1.66$  BCR10 and  $SD=0.2$ ) and the most uncomfortable reported were *orthogonal* gestures ( $M=2.35$  BCR10,  $SD=0.38$ ). It is worth mentioning that due to the number of trials (192) during the experiment, fatigue could potentially affect participants' performance. However, as shown in **Figure 3.4(c)**, the maximum score of effort was about 3.2 (in a scale from 0 to 10) suggesting that although we could observe differences in effort (e.g., *orthogonal* gestures were more uncomfortable), participants gave generally low scores in effort and therefore we considered unlikely that these low scores represent a negative effect on participants' performance during the experiment.



**Figure 3.5** Average time (a) and accuracy (b) for the first and second gesture. Error bars represent standard error of mean.

*Typing speed of 1 second (C2)*: The effects of typing speed on gesture time (**Figure 3.5(a)**) and accuracy (**Figure 3.5(b)**) were also analysed. This revealed the first gesture ( $M=0.783s$ ;  $SD=0.119s$ ) is significantly shorter than the second one ( $M=0.843s$ ;



$SD=0.109s$ ), and also more accurate ( $p=0.012$ ). Using an input speed of 0.8s users barely could keep up with the input speed (first gesture  $>0.8s$ , accuracy significantly smaller than input at 1.2s ( $p<0.001$ )). It is interesting how users (even if allowed more time) did not take more than 0.97s to perform each gesture. No significant differences were found for typing speeds of 1s or 1.2s. Thus, we included typing speed of 1s (**C2**) as a low-level constraint (i.e., fastest speed allowing sustained typing).

### 3.4.3 Designer's Workshop.

After obtaining the relevant low-level factors, we carried out a workshop with interaction designers, as a mean to identify the design rationale they use in producing their mappings. We motivated the workshop around the concept of gestural text-entry, a challenging context forcing them to explore the topic in depth.

We recruited four UX designers (no specific expertise on text-entry) from the University of Bristol (other than where the main study was conducted), to produce a design scheme for the system. The workshop session lasted four hours. To encourage a broad perspective towards the design of an effective *interactive dialogue*, designers were encouraged to think about these four questions: *How to map gestures with letters? What is a good interface layout? What feedback elements are required? Is the operation easy to remember?* The workshop was kept open-ended to encourage creative thinking, but one researcher stayed in the room, to answer designers' questions. It must be noted that the quantitative results from (i) (e.g., speed, accuracy) were only provided if and when specifically requested by designers, to not bias their thoughts. This request was made one hour after the workshop started when designers wanted to identify the speed ratios per gesture combination to be considered in the mapping design.

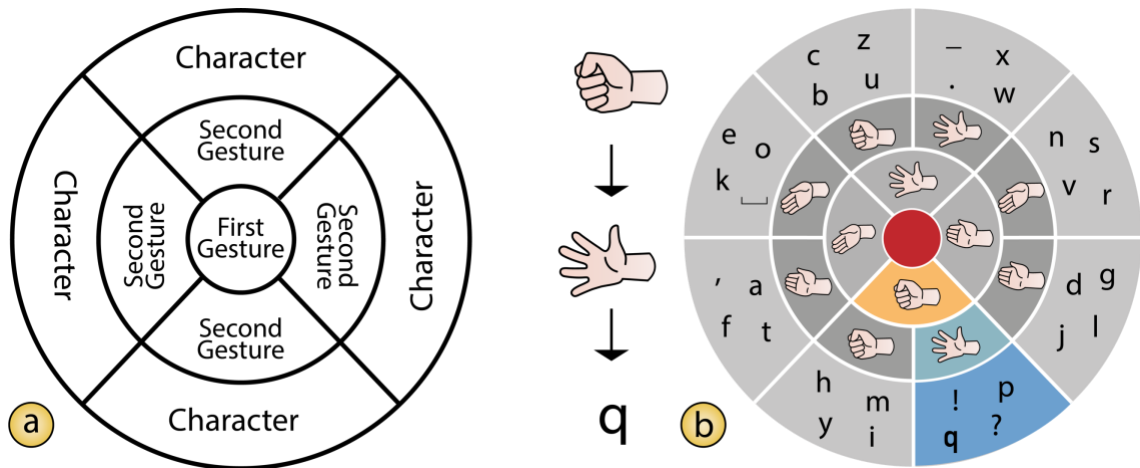
At the beginning of the workshop, designers considered using chained gestures right away. Three-chain gestures were soon discarded by designers, due to their high cognitive load (too many potential gestures to remember) and discomfort (*orthogonal* and *opposite* gestures). Thus, they limited their search to 2 step-chain gestures (**C1**) and a predictive text entry. This used 8 categories, mapping 4 letters to each gesture/category and addressing 32 characters: the 26 letters from the English alphabet and the 6 most common punctuation characters: space, period, comma, question mark, exclamation mark and hyphen). They also felt inclined to explore alternatives beyond the constraints defined (such as using both hands or using continuous gestures, using the duration of the gesture as a variable). At the end of the workshop, designers were asked to present their interface

layout and to reflect on it, as a way to verbalize their rationale. In the next subsection, we report these observations as *high-level factors* and *constraints*.

From designers' rationale to factors and constraints. Designers soon got interested in the time (**F1**) and accuracy (**F2**) of each gesture and experimented with the level of comfort (**F3**) afforded by each gesture by performing them casually. They considered the WI gesture to be the most 'natural' gesture, and WO as the least comfortable. They also found the F and S gestures hard to perform. Designers also became interested in the frequency of using each letter, using the ENRON corpus (Klimt et al., 2004) to inform this aspect.

At the end of the workshop, they presented their proposed interface design (see **Figure 3.6(a)**), reflecting both the appropriate interface design and the way the *interactive dialogue* should work. The UI layout consisted of several concentric circles, working as a decision tree with choices at each node. Users would identify the target letter in the external level/ring and then follow the path through the ring from the inside out, performing the gestures to reach the chosen letter. The interface should highlight the rings, as gestures are recognized, e.g., **Figure 3.6(b)**, shows Fist + Spread gestures used to type 'q', and feedback displayed.

The final scheme presented reflected aspects of their rationale (*high-level factors*), highly relevant for our approach. For instance, they attempted to maximize the usage of WI (**F4**), while avoiding WO (**F5**) and S gestures (**F6**). They also found the use of *orthogonal* gestures very uncomfortable and suggested avoiding them (**C3**).



**Figure 3.6 Interface layout: a) proposed by designers; b) Final design using their factors and our search method. Typing a “q” requires to perform the chain gesture fist (F) - spread (S).**

As a second major concern, designers also attempted to reduce the cognitive load of the mapping, by applying several heuristics. For instance, they suggested to keep all vowels clustered together (in two categories only) (**F7**). They also placed alphabetically adjacent letters in the same categories (e.g., “abcd”), which was considered as a relevant factor (**F8**). These techniques were meant to facilitate users' ability to remember the layout.

Designers also tried to assign the comfortable and fast gestures to the most frequent characters. They attempted to build a mapping solving the problem in an optimal way, and including all identified factors. However, they failed to find a clear candidate mapping, illustrating the challenge designers face when addressing large solution spaces.

#### 3.4.4 Formalization & Optimization.

We used the *constraints* (C1-C3) and *factors* (F1-F8) identified in the previous stages to refine our definition of the problem and to formalize the description of our candidate mappings. Due to our constraints, we limited our search to 2-step chain gestures (C1), with typing speed 1s (C2) and used only "opposite" and "repeated" gestures (C3), resulting in only 8 possible gesture chains (see **Figure 3.3**).

Each factor was formalized (quantified), with the common criteria that lower values represent a better mapping. Let  $D$  be our dictionary (we use the ENRON database (Klimt et al., 2004), with duplicates to represent word frequency). Let  $W$  be a word and  $L$  a letter. Let  $Time(L)$ ,  $Accuracy(L)$  and  $Exertion(L)$  be the meantime, accuracy and effort (i.e., inverse of comfort) of the gesture associated with letter  $L$ , as measured from our quantitative studies from (i).

*Time factor (F1)*. This factor favours fast typing speeds, by quantifying the "average time to input a letter according to our dictionary".

$$F1(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{Time}{|D||W|} \quad (3.2)$$

*Accuracy factor (F2)*. This factor enforces mappings with gestures of high accuracy recognition, by quantifying the “probability to make one (or more) errors in a word”.

$$F2(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{1-Accuracy(L)}{|D|} \quad (3.3)$$

*Comfort factor (F3)*. This factor measures the “amount of exertion required to input a letter”, to minimize effort.

$$F3(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{Exertion(L)}{|D||W|} \quad (3.4)$$

Wave-in factor (**F4**). This factor encourages the use of WI gesture, considered comfortable by designers. This factor computes "the average density of non-WI gestures per letter".

$$F4(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{\text{isNot } Wi(L)}{|D||W|} \quad (3.5)$$

Wave-out factor (**F5**). This factor discourages the use of WO gesture, as it was considered less comfortable. Particularly, it quantifies "average density of WO gestures per letter".

$$F5(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{\text{is } Wo(L)}{|D||W|} \quad (3.6)$$

Spread factor (**F6**). This factor penalizes the use of S gestures, as they were considered less comfortable. This factor computes the "average density of S gestures per letter".

$$F6(M) = \sum W \text{ in } D \sum L \text{ in } W \frac{\text{is } S(L)}{|D||W|} \quad (3.7)$$

Vowels factor (**F7**). This factor counts the "number of categories containing vowels", to favour vowels being grouped in a few categories.

$$F7(M) = \max(|V|), V \subset C / \forall c \in V, \{a, e, i, o, u\} \cap c \neq \emptyset \quad (3.8)$$

*Consecutive factor (F8)*. This factor benefits mappings where letters are assigned to categories in a consecutive order. Thus, it measures the "number of non-consecutive (NC) letter per category (C)".

$$F8(M) = \frac{NC(C[0],C[1]) + NC(C[1],C[2]) + NC(C[2],C[3])}{3} \quad (3.9)$$

*Determining the weight of each factor and optimization.* Each factor was normalized to a  $[0, 1)$  range, as in **Table 3.1**. This allows the relevance of each factor to be assessed in terms of weight alone (and not according to the factor's scale). Constants  $s_w$  and  $l_w$  represent the length of the shortest and longest words in  $D$ , respectively;  $m_t$  and  $M_t$  stand for the minimum and maximum gesture times, and  $m_a$  and  $M_a$  stand for the minimum and maximum gesture accuracy respectively. Weights were then determined based on the designers' insight. It must be noted that this was the interpretation of the research team (i.e., two transcribing and cross-validating notes from the experiment, and two translating them into the weights described in **Table 3.1**), as we had no further access to the designers involved in (ii).

	F1	F2	F3	F4	F5	F6	F7	F8
Min	$ s_w  \cdot m_t$	$ s_w  \cdot (1 - m_a)$	0.125	0	0	0	2	0
Max	$ l_w  \cdot M_t$	$ l_w  \cdot (M_a)$	1.75	1	1	1	5	1
Ki	0.35	0.20	0.1	0.05	0.05	0.05	0.1	0.1

**Table 3.1** Factors used use for  $M_{Des}$  (our proposed mapping), ranges and weights (ki).

We used these weights (cost function as described by Eq(3.1)) and simulated annealing (SA) (Kirkpatrick et al., 1983) to find the optimum mapping. Initially, letters were randomly assigned to the 8 categories (only "opposite" and "repeated" gestures, see **Figure 3.3**) and neighbour states were computed by permutation of single letters between two random categories (diameter=32). Transition acceptance between states follows the traditional method by Kirkpatrick (Kirkpatrick et al., 1983). Cooling schedule was empirically tuned with  $N_s=20$  step adjustments per temperature step,  $N_t = 7$  temperatures steps per temperature change,  $R_t = 0.85$  (Cooling factor). The initial temperature was set in  $T(0)=180$ . The final mapping is shown in **Figure 3.7(a-c)**.

Given the designers constraints (no *Orthogonal* gestures), the solution space was limited to  $\binom{32}{4} = 35960$  mappings and a full search would have been feasible. However, this was not feasible for the pure computational solutions we compared against (larger solution space), and we used the same schedule to aid fairness in comparison.

#### 3.4.5 Computing alternative approaches.

Some of the factors and observations made by designers were hard to justify purely looking at the data. The categories identified (*Repeat*, *Opposite* and *Orthogonal*) show weak differences and, given any performance metric, all of them have gestures both well above and below the sample mean. Even in the case of time per gesture (clearer distinctive behaviour for *Opposite*), the use of clustering techniques would not result on the categories identified.

Picking specific data could seem to back up the designers' insight. For instance, *WI+WI* was the most comfortable gesture ( $M=1.15$  Borg CR10 Scale –BCR10) and *WO+S* as the least comfortable ( $M=3.15$  BCR10), followed by *S+WO* ( $M =2.6$  BCR10). While *WO+WI* resulted the fastest 2-step chain gesture ( $M=1.947s$ ,  $SD=0.228$ ), *WI+WI* was second fastest ( $M=1.949s$ ,  $SD=0.242$ ), the most accurate ( $M=95.13\%$ ,  $SD=13.73$ ) and the most comfortable gesture performed ( $M=1.15$  BCR10), whilst *WO+S* the least comfortable ( $M=3.15$  BCR10).

These point observations could support designers' factors **F3** and **F4**, but observational bias and the limited size of the sample would make for weak evidence. This was found worrying, as it could point towards a weak ability of the designers to analyse the complexity of the problem. On the other hand, factors could also reflect designers' *tacit knowledge*, that is, understanding of complex mechanics of the task which were difficult to articulate, but still relevant.

Thus, we decided to compare the designer's guided solution against six naïve computational solutions, not considering designers' *high-level factors* and *constraints* (e.g., 8 categories used to allow comparison, but not constrained to *Repeat* and *Opposite* gestures alone). These naïve solutions will both help us assess the added value introduced by feeding the designers' insight into the optimization method; and also challenge their decisions/constraints.

These six solutions were generated as a combination of two elements: **a)** the training dataset: the Enron (E) dataset (Klimt et al., 2004); the most common Digraphs (D) in English language (Bowman et al., 2004), and a combination of both (E+D); and **b)** the cost functions: two were defined, one assessing time per gesture (factor **F1**) and another one assessing accuracy (**F2**). e.g., **M\_C1** represents the mapping obtained with the best *Accuracy* assessed by Digraphs dataset. For each of the six combinations, we generated all the possible subsets of 8 gestures (from the 16 different 2-step gestures possible) and used Simulated Annealing to compute the best letter combinations. We explored  $\binom{16}{8} \cdot \binom{32}{4} \cdot 6 = \sim 2.8$  billion possible mappings, with **Figure 3.1** showing the best mapping for each of the 6 naïve cost functions.

### 3.5 Analytical and Summative Evaluation

**Figure 3.1** shows histograms for all possible mappings according to our seven metrics (the naïve computational metrics (a-f) and designer-led (g)). The best mappings per metric are also highlighted (as colour bars) in the remaining histograms, for comparison. **Table 3.2** shows this information in a numerical format. The best results for Accuracy mappings (i.e. **M<sub>c1</sub>**, **M<sub>c3</sub>** and **M<sub>c5</sub>**) was **M<sub>c5</sub>** (best average percentile across the 6 naïve functions, within its category), while the selected mapping for Speed (i.e. **M<sub>c2</sub>**, **M<sub>c4</sub>** and **M<sub>c6</sub>**) was **M<sub>c4</sub>**. For clarity, during the comparative evaluation, we will refer to these as time (**M<sub>Ti</sub>**), and accuracy mappings (**M<sub>Acc</sub>**), instead of (**M<sub>c4</sub>** and **M<sub>c5</sub>**).

Cost/Mapping	M_C1	M_C2	M_C3	M_C4	M_C5	M_C6	M_Des	Average	SD
<b>C1-A<sub>Acc</sub></b>	0	33	5	3	1	31	33	0.304	0.077
<b>C2-A<sub>Sp</sub></b>	21	0	65	1	45	1	67	1.147	0.052
<b>C3-E<sub>Acc</sub></b>	4	100	0	11	1	29	5	0.238	0.066
<b>C4-E<sub>Sp</sub></b>	81	50	45	0	60	1	40	1.077	0.064
<b>C5-D+E<sub>Acc</sub></b>	1	80	1	3	0	20	10	0.587	0.014
<b>C6-D+E<sub>Sp</sub></b>	49	4	42	1	40	0	41	2.258	0.110
<b>D-D+E<sub>Mix</sub></b>	10	40	1	1	1	1	0	0.949	0.025

**Table 3.2 Numerical mappings’ performance: The percentile per mapping (0 to 100) across the seven cost functions (CF) used in the optimization process. On the right columns, AVG and SD for the data per CF condition are shown. The best mappings for speed (M\_C4) and accuracy (M\_C5) are highlighted in green while our proposed mapping (M\_Des) is highlighted in blue.**

It was also interesting to see how the designers-led layout ( $M_{Des}$ ), rated against the other mappings. While computational mappings consistently scored well using the designers’ cost function (see last row), the designers mapping scored much more mediocre results (see column  $M_{Des}$ ), being usually in fourth or fifth position (or even last) among the mappings considered.

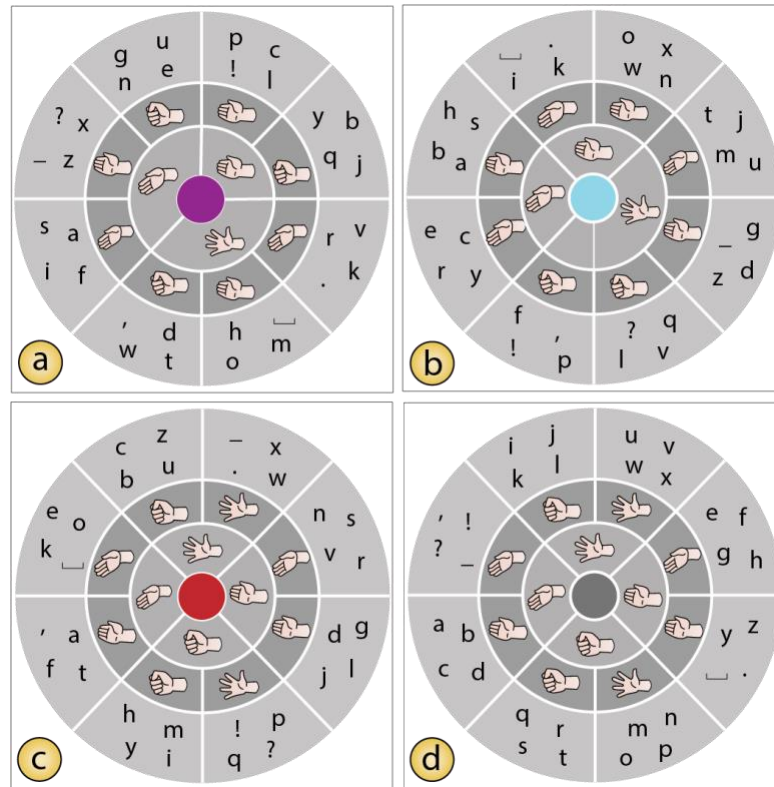
We then carried out a user study to evaluate the performance of the generated mappings:  $M_{Ti}$ ,  $M_{Acc}$  and  $M_{Des}$ . We added one additional mapping for text-entry i.e. a simple alphabetical distribution ( $M_{Abc}$ ) shown in **Figure 3.7(d)**, as a baseline comparison (minimum cognitive load, not optimized).

### 3.5.1 Experiment Setup

At the beginning of the session, we calibrated the Myo for each individual participant. Subsequently each mapping was shown on screen with its different layout and letter distribution (see **Figure 3.7**). Participants were then instructed to “type” a sentence shown above the circle by performing the specific chain of gestures (i.e., identifying the two gestures they need to perform to select a given letter). The system included feedback cues i.e., visual highlights in the category selected at each step (see **Figure 3.6(b)**), and auditory effects.

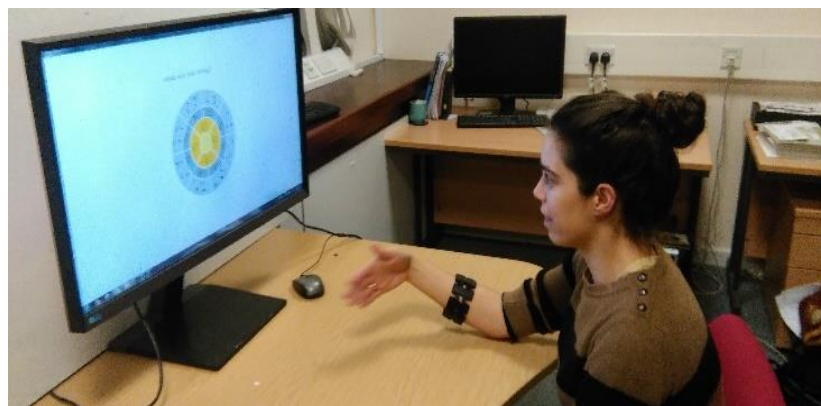
Participants were allowed to practice the chain gestures in a training stage to complete 4 sentences before each block, in order to get familiar with the layouts. Participants performed 4 blocks of 3 sentences each, completing 28 sentences in total (700 letters/gesture chains). The sentences in the blocks had from 4 to 6 words, and 4 to 6 letters per word, being selected by using the Levenshtein algorithm (Haldar et al., 2011) to compute representative sets of sentences from our dictionary.





**Figure 3.7 Final gesture mappings: a) Gesture mappings of time factor ( $M_{Ti}$ ), (b) accuracy factor ( $M_{Acc}$ ), (c) mixed mapping according to designers' factors ( $M_{Des}$ ) and d) alphabetical gesture mapping ( $M_{Abc}$ ).**

The full experiment duration was 45min. Similarly, as described in the first study, each block was designed to be completed in about 8min giving participants a 3min break between blocks to avoid fatigue. Moreover, since *orthogonal* gestures (the most uncomfortable gestures found in the first study and rated on average  $\sim 3.2$  in a scale from 0 to 10) were not employed in this study, we considered unlikely that fatigue negatively affects participants' performance during the experiment. We counterbalanced the order of the sets (i.e., sentences) and mappings using a  $4 \times 4$  Latin Square design. **Figure 3.8** shows our experimental setup.



**Figure 3.8 Experimental setup for the typing task.**



The system collected the time per letter and error rate automatically. User-satisfaction questionnaires after each block (mapping), collected information about typing comfort and how easy each it was to remember each mapping. Finally, at the end of the experiment, participants also chose their favourite mapping according to 4 aspects (*easy to type, comfort, speed and easy to remember*). Sixteen right-handed participants took part in the experiment (4 Females, average age of 29), which was approved by the local ethics board. The experiment duration was 45min, including 3 min breaks between conditions. The recruitment criteria were the same as in the first experiment.

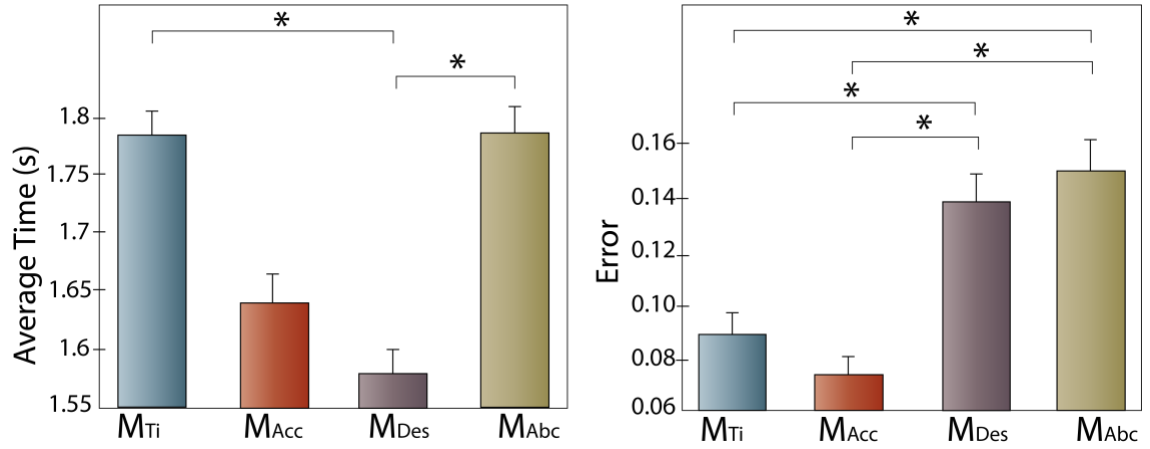
### 3.5.2 Analysis of Results

An a priori statistical power analysis was performed for sample size estimation in G\*Power. Running a power analysis on a repeated measures ANOVA mapping conditions (i.e.,  $M_{Ti}$ ,  $M_{Acc}$ ,  $M_{Des}$  and  $M_{Abc}$ ), repeated 28 times corresponding to the 28 sentences on the experiment), a power of 0.95, an alpha level of 0.05, and a medium effect size ( $F=0.196$ ,  $\eta^2=0.037$ , critical  $F=1.1$ ), requires a sample size of approximately 8 participants. Thus, our proposed sample of sixteen participants was adequate for the purposes of this study.

A Repeated Measure ANOVA was conducted to compare the effect of the four type of mappings ( $M_{Ti}$  vs  $M_{Acc}$  vs  $M_{Des}$  vs  $M_{Abc}$ ) on the time of chain of gestures. Results revealed a significant effect on the average time,  $F_{(3,45)}=25.82$ ,  $p<.001$  depending on the type of mapping, with the designers' mapping providing best results. Post-hoc comparisons using Bonferroni correction showed statistically significant differences in time, specifically between  $M_{Des}$  ( $M=1.577s$ ,  $SD=0.622s$ ) compared to  $M_{Abc}$  ( $M=1.785s$ ,  $SD=0.674s$ ;  $p<0.001$ ), but also  $M_{Des}$  and  $M_{Ti}$  ( $M=1.782s$ ,  $SD=0.653s$ ;  $p<0.001$ ). No such difference was found compared to  $M_{Acc}$  ( $M=1.64s$ ,  $SD=0.71s$ ),  $p=0.279$ . Surprisingly,  $M_{Ti}$  did not provide the best results for time, which seems to indicate it failed to capture the complexity of the typing task.

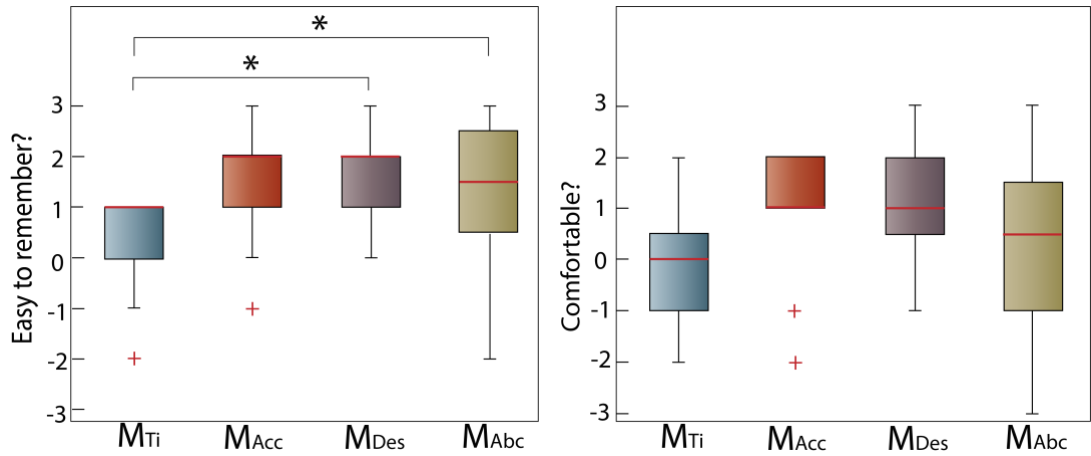
The average error per mapping was small for all conditions. As expected,  $M_{Acc}$  got the lowest error score as it was computed to minimize errors. A Repeated Measure ANOVA test showed a significant effect of the type of mapping ( $M_{Ti}$  vs  $M_{Acc}$  vs  $M_{Des}$  vs  $M_{Abc}$ ) on the number of errors  $F_{(3,45)}=7.71$ ,  $p<.001$ ,  $\eta^2=0.009$ . Post-hoc comparisons showed statistically significant differences for errors, specifically between  $M_{Acc}$  ( $M=0.072$  errors,  $SD=0.293$  errors) compared to  $M_{Des}$  ( $M=0.139$  errors,  $SD=0.444$  errors),  $p=0.001$  and

$M_{Abc}$  ( $M=0.149$  errors,  $SD=0.520$  errors),  $p=0.001$ ; but no such difference was found compared to  $M_{Ti}$  ( $M=0.087$  errors,  $SD=0.369$  errors),  $p=1$ .



**Figure 3.9 Comparison of performance: Scatter plot of AVG gesture time (Left) and errors (Right) per mapping. Bars represent standard error of mean.**

Additionally, we found a significant difference in  $M_{Ti}$  compared to  $M_{Des}$ , and  $M_{Abc}$ ,  $p \leq 0.035$ . These results suggest that  $M_{Acc}$  and  $M_{Ti}$  produced the lowest number of errors when participants performed the gesture chains to “type” the sentences.



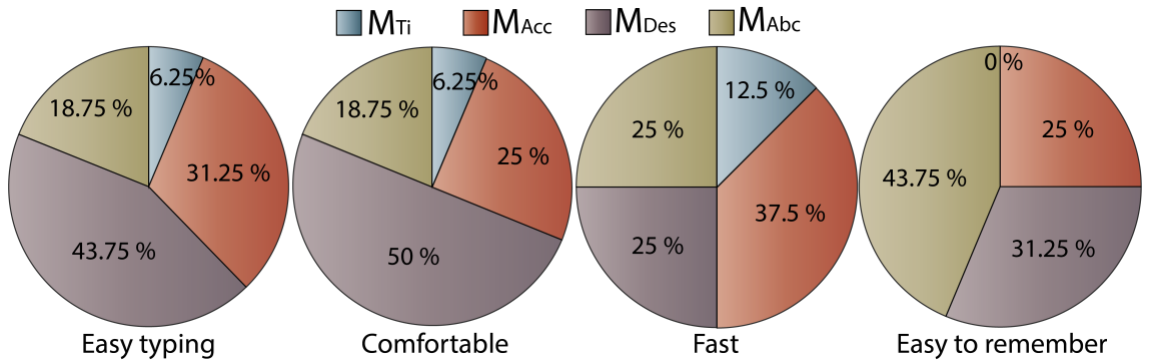
**Figure 3.10 User Experience results: Box plots for rememberability (left) and comfort (right) per mapping. Horizontal red bars and boxes represent medians and IQRs. Whiskers stretch to points within median  $\pm 1.5$  IQR. Outliers shown as single red crosses.**

**Figure 3.10** shows the score given by participants (using a scale from -3 to 3) after each block in relation to how easy to remember the mapping was (left) and how comfortable they felt while “typing” (right). A Repeated Measure ANOVA test showed a significant effect of the mapping used on participants’ scores of rememberability  $F_{(3,33)}=11.7$ ,  $p < .001$ ,  $\eta^2 = 0.723$ . Post-hoc comparisons using Bonferroni correction showed that there is a statistically significant difference between  $M_{Ti}$  ( $M=0.67$ ,  $SD=0.98$ ) and  $M_{Des}$

( $M=1.67$ ,  $SD=0.88$ ,  $p<0.05$ ), and between  $M_{Ti}$  and  $M_{Abc}$  ( $M=2.17$ ,  $SD=0.71$ ,  $p<0.01$ ). However, we found no statistically significant effect of the mapping used on participants' scores of comfort  $F_{(3,33)}=1.39$ ,  $p=0.26$ ,  $\eta^2=0.112$ , although higher mean scores are observed for  $M_{Acc}$  and  $M_{Des}$  as shown in **Figure 3.10**(right).

Particularly, when participants were explicitly asked which mapping they preferred in terms of ease, comfort, speed and rememberability (see **Figure 3.11**), most of the participants reported  $M_{Des}$  as the most comfortable (50%) and easiest to type mapping (43.75%), followed by  $M_{Acc}$  (31.25% and 25%, respectively).

Interestingly, although  $M_{Des}$  allowed for faster typing as shown in the performance results (see **Figure 3.9**, left),  $M_{Acc}$  was reported as faster by most of participants (37.5%). Finally, as expected, participants also reported  $M_{Abc}$ , as the easiest to remember (43.75%), followed by  $M_{Des}$  (31.25%).



**Figure 3.11 Percentage of participants that preferred each mapping: ( $M_{Ti}$ ,  $M_{Acc}$ ,  $M_{Des}$  and  $M_{Abc}$ ) regarding their task experience (ease typing, comfort, speed and ease to remember).**

### 3.6 Discussion

Our results seem to indicate the designer-led semi-automatic mapping  $M_{Des}$  provided better results in terms of time, comfort and users' preference (i.e., most of participants preferred  $M_{Des}$  in terms of ease and comfort) when compared to the remaining mappings. It consistently appeared as the best or second-best option, only performing worse in terms of accuracy, where very small differences (effect size) were present among mappings. This suggests that users preferred the mappings created by the combinations of experts' knowledge (proposed weights for  $M_{Des}$ ) and the computational optimization.

This might reflect the difficulty to model all aspects related to interaction using only low-level factors, and how these might be misleading when the complexity of the task increases. Even for our naïve cost functions,  $M_{Ti}$  did not actually lead to faster typing

speeds; and they also failed to predict the performance of  $M_{Des}$  (expected to be poor, as shown in **Table 3.2**), even for the specific factors (i.e., time) they measured.

The results also highlight the value of designers' higher-level insight, even if it cannot be directly justified from data. For instance, the categories identified (*Orthogonal*, *Repeat*, *Opposite*) guided constraint **C3**, but they could not be identified from clustering techniques. During the workshop, we pointed out that the *high-level factors* **F4**, **F5** and **F6** were already covered by *low-level* ones, but designers still decided to keep them. We understand these reflect *tacit knowledge* which, even if hard to verbalize/rationalize, was still relevant to the task. The results obtained by the designers' mapping should highlight the relevance of such designers' insight (i.e., high-level factors identified), but it also illustrates the value of our hybrid approach, exploiting computational methods to keep this human knowledge in the optimization loop.

The resources required for both the designers' workshops and the brute-force exploration of alternative mappings must also be considered. The full search to create our alternative mappings (2.7 billion combinations explored, for the 6 alternatives) required 5 standard desktop machines running over 5 days (development costs for software not considered). In comparison, the designers' feedback was gathered during a single workshop of 4 hours and still managed to identify relevant *high-level factors*, *constraints*, and provided good results for the final mapping. This seems to indicate designers' involvement can be easily justified, producing relevant input to underlying computational approaches and potentially reducing development costs.

Finally, our use-case must be considered as an illustrative example of our approach, rather than an exemplar text-entry system. Text entry systems can leverage extensive task-specific knowledge (e.g., digraph transitions, predictive models, etc.), which can allow defining effective mappings even from low-level factors. Instead, our case study provides an example that is generalizable to a broader spectrum of applications using gestural interaction; illustrates the challenges related to creating complex *interactive dialogues* from low-level factors; and highlights the benefits related to designers' insight into the process. Our approach can be seen as a guide to develop useful tools from computational approaches to help designers not only on the low-level factor identification, but also in the exploration of big search spaces while keeping control over the main steps of the process, such as i) the high-level factors definition to better reflect their insight about the problem and ii) the relevance that each of the factors has over the optimization.

### 3.7 Conclusion

In this chapter we presented an approach for semi-automatic generation of gesture mappings for devices with low gestural resolution. Our approach consists of quantifying observable *low-level factors* such as individual gesture error rates, speed and accuracy; and identifying how designers weigh different factors to create a weighted cost function that is optimized to find the gesture set and its mappings to tasks. Comparing the results of our mapping with mappings obtained from other naïvely constructed cost functions shows that overall users perform consistently well with our mapping in terms of speed, comfort and memorability. These results highlight the value of our approach, as a tool to guide the designer led computational approach to generate complex mappings. This approach should not stand as a replacement for traditional HCI methods, but as a tool to help such iterative processes to converge faster towards satisfying solutions.

It is worth mentioning that with the proposed generation of mappings, we also enabled the use of low input resolution devices to achieve complex tasks (as the case of study of this chapter). This may facilitate in some way, the coupling between hardware input and ITes (even with different input/output resolutions).

## Chapter 4

# Erg-O: Ergonomic Optimization of Immersive Virtual Environments

Following our thesis structure, in this chapter we explore the second stage of interaction proposed by (Molina-Masso et al., 2008) shown in **Figure 2.4** (i.e., ITes). We particularly focus on *object manipulation* as it has been identified as one of the primary VR interaction tasks (Boyd et al., 1999). Here we discuss user fatigue common during mid-air *object manipulation* in VR. To address this issue and following our general approach, we designed a tailored ITe to achieve more comfortable *object manipulation* taking into consideration *human factors*. We combined ergonomic evaluations and computational optimization methods to generate a comfort mapping that represents the lowest effort areas for *object manipulation* within the user's arms reach. This approach aims at exploiting the user capabilities to interact in 3D unconstrained spaces (i.e., mid-air interaction) but at the same time, addressing human limitations (fatigue) inherent within mid-air interactions in 3DUIs.

### 4.1 Introduction

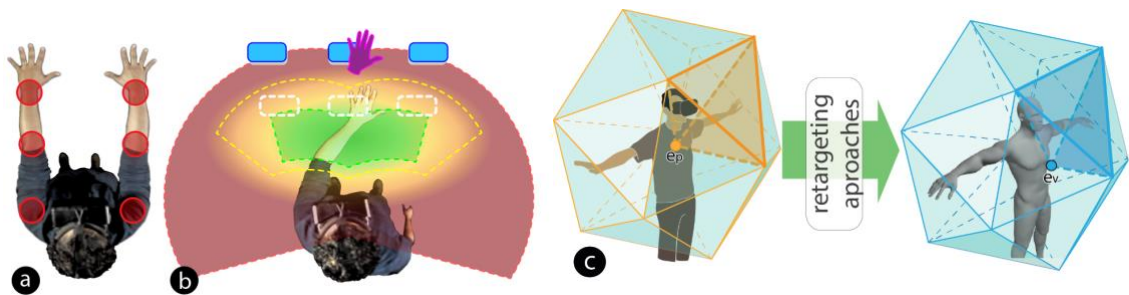
Recent market studies foresee VR will become mainstream, reaching a \$62 billion market by 2025 (Ott et al., 2015). The rise of commercial VR devices, tracking technologies and 3D graphics have enabled increasingly compelling VR systems, not only in displaying realistic content but also allowing more natural interactions (Steuer, 1992) and better feeling of presence (Sanchez-Vives et al., 2005). Beyond entertainment, training environments can easily take advantage of this (e.g., flight or surgery simulators (Satava, 1993; Seymour et al., 2002)), as users are allowed to interact with the virtual environment

(VE) in much the same way as they would do in reality, softening the learning process (Bricken, 1991).

However, such natural VR interaction often involves large body motions (mainly affecting upper limbs) (Wachs et al., 2011), which can result in fatigue and discomfort (Hincapié-Ramos et al., 2014) (see **Figure 4.1(a)**). This is especially true for demanding (i.e., complex or repetitive) tasks, or gaming activities for long periods.

One possible solution is to place the interactive elements (e.g., buttons, menus) at ergonomically comfortable positions, using ergonomic evaluation metrics such as RULA (McAtamney et al., 1993) or Jack (Badler et al., 1993). This can be useful for in-game menus or in scenarios where the VR designer is free to pick the location of the interactive elements around the user.

Unfortunately, such ergonomic relocation might not be applicable to a pilot cockpit, or training scenarios where the virtual object resembles a real one, and interactive parts cannot be relocated. Manipulation techniques, such as Go-Go (Poupyrev et al., 1996), might allow users to reach distant objects, while keeping arms in closer, more comfortable positions for the user (i.e., avoid overstretching of the arms). However, this technique loosens the egocentric manipulation metaphor (i.e., virtual hand), reduces precision at longer distances (Flasar, 2001), affects the feeling of body ownership (Lopez et al., 2008) and can be undesirable for training/simulation scenarios where the user needs to be aware of the actual limits of his interaction space (i.e., what she/he will actually be able to reach/do in the real situation)(Richardson et al., 1999; Dünser et al., 2006).



**Figure 4.1 Erg-O approach: (a) VR involves interactions with upper limbs, which can lead to discomfort (b) Out approach retains visual objects in their location, but users can reach them from more comfortable positions (c). Our approach is based on defining two space partitioning trees, and using optimization approaches to look for most comfortable retargetings (visual to physical positions).**

Our proposed solution is to get benefit of the dominance of human visual system over the proprioceptive system. We retain the visual position of the elements in the VE, but allow

users to reach them from more ergonomic physical positions (**Figure 4.1(b)**). This is possible as changes in position of only a few cm can increase comfort significantly.

We first contribute a manipulation technique that allows such ergonomic retargeting for a variable number of interactive elements within the user's arm reach. Our solution wraps the interactive space around the user, ensuring that: **a)** the virtual hand reaches the visual location of the interactive element, when the physical hand reaches the retargeted physical location of the element (**Figure 4.1(b)**); **b)** the technique works in an open ended fashion, not needing prior knowledge about the element the user wants to reach at each point; and **c)** any other point within the user's arm reach is still reachable, with continuity of interaction even when reaching between interactive elements.

We combine our manipulation technique with optimization methods, to enable online computation of optimum *retargeting mappings* (i.e., most ergonomic retargeted position to interact with the visual representation of each interactive element). We describe two example optimization strategies to obtain such mappings (*Spatially Consistent* ( $S\_R$ ) and *Ergonomic* ( $E\_R$ )) and report the results from a user study with 12 participants, comparing  $S\_R$  and  $E\_R$  approaches to natural virtual hand interaction.

Our results show that participants' comfort was improved according to quantitative data (RULA score) as well as subjective judgement in retargeting conditions ( $S\_R$  and  $E\_R$ ) compared with the natural ( $N$ ) condition (one-to-one mapping without retargeting). Additionally, we found that execution time was lower in  $S\_R$  and  $E\_R$  conditions compared with  $N$  condition. These results illustrate the benefits that the multi-object retargeting enabled by ERG-O can provide for a general VR system using virtual hand interaction. We finish the chapter reflecting on how the technique can also be applied for other application scenarios, such as rehabilitation or reinforcing spatial skill training for patients with cerebral palsy.

Finally, it is worth mentioning that the approach in this chapter is focussed on scenarios where virtual elements in the scene cannot be visually modified as position changes negatively affect space awareness of the user. For instance, in virtual pilot training, the position of the elements in a cockpit cannot be changed as the pilot must generate a mental model that should match with the layout of a real control panel. For this reason, in order to facilitate this mental mapping and decrease the movement effort during virtual training, we proposed our redirection technique base on an automatic ergonomic evaluation.



## 4.2 Related Work

Our technique can be categorized as an egocentric virtual hand metaphor according to VR manipulation taxonomies ([Flasar, 2001](#)). To better appreciate our contribution, our review is focused on two main areas: (1) visual dominance and spatial redirecting; and (2) ergonomic assessment.

### 4.2.1 Visual Dominance and Spatial Redirecting

Visual dominance refers to the tendency of visual information to determine what is perceived when conflicting information is perceived through the visual channel and any other modality ([Colman, 2015](#)).

This effect has been extensively exploited in VR ([Burns et al., 2005](#)), with best known applications for navigation techniques such as redirected walking, or to avoid visual penetration of the virtual hand inside solid objects (e.g., rubber-band virtual hand ([Burns et al., 2005](#))). However visual dominance can also influence the way in which we perceive our own body, such as having a bigger belly or even having a child's body ([Normand et al., 2011](#); [Kilteni et al., 2012](#); [Banakou et al., 2013](#)). When combined with synchronized multisensorial stimuli, it can even be used to induce illusions of executing actions, such as speaking ([Banakou et al., 2014](#)) or walking ([Kokkinara et al., 2016](#)).

Closer to our approach, visual dominance has also been studied in the context of hand interaction. Burns et al. ([Burns et al., 2005](#)) found very strong dominance of visual over proprioceptive perception when no tactile feedback is provided. This allowed for up to 20 cm just noticeable differences (JND) between the real and virtual hand location, before becoming noticeable (75% recognition rate), even if users were aware that a mismatch could become present.

This mismatch threshold is significantly reduced if vibrotactile cues are introduced to reinforce proprioception or when other body parts are involved. Lee et al. ([Lee et al., 2015](#)), reported JND thresholds of 5.2 cm when cutaneous haptic feedback (normal and shear forces) was applied to the fingertip. Matsuoka et al. ([Matsuoka et al., 2002](#)) report average JND thresholds of 3.2 cm for finger flexion when force feedback is applied. Direction of forces ([Barbagli et al., 2006](#)) or the curvature of the physical props ([Robles-De-La-Torre et al., 2001](#)) can also influence these thresholds.

This knowledge has allowed the development of various redirection techniques for manipulation. Haptic retargeting ([Azmandian et al., 2016](#)) and Sparse Haptic Proxy

(Cheng et al., 2017) create the illusion of touching several virtual objects. Unlike Erg-O, the target of interaction must be known a priori and it only applies to stream-lined interaction (i.e., hand at a rest position, then touch the target object), not allowing free hand movements. Valkov et al. proposed a technique using the display surface of a stereoscopic flat display as the passive haptic prop for shallow 3D interaction (Valkov et al., 2014). Redirected touching (Kohli et al., 2012) uses a flat board to induce the feeling of touching rotated objects. Unlike ERG-O, these techniques are usually limited to a single point of interaction (i.e., a finger) and require previous knowledge on the target of the interaction.

Leveraging thin-plate spline warping (Boring et al., 2009), approaches have been reported that allow mapping point interactions (e.g., fingers, surgery tools) to passive, non-flat surfaces of known geometry (Kohli, 2010; Ban et al., 2012; Spillmann et al., 2013). However, ERG-O is the first VR manipulation technique to tackle redirection for the whole interactive space around the user, not being limited to single points or surfaces and operating in an open-ended fashion (i.e., target of interaction not known a priori).

#### 4.2.2 Ergonomics

Ergonomic assessment has been extensively used to assess risks in workspaces, but also to evaluate interaction within HCI. These methods can be divided into: self-report, observational methods and direct measurement (Burdorf et al., 1991).

Self-report methods (e.g., NASA-TLX (Bustamante et al., 2008) or the Borg CR10 scales (Borg, 1998)) usually involve questionnaires, ranked by using Likert-scales. These methods, however, do not allow for online assessment (while the task is carried out), and the need to rate difficultly quantifiable parameters (e.g., workload) can compromise reliability of the results (Wiktorin et al., 1993).

Observational methods and direct measurement allow for online assessments, and the development of marker-less sensing techniques is slowly removing this distinction. Previous observational methods such as RULA (McAtamney et al., 1993) or Jack (Badler et al., 1993) can now be directly measured using nonintrusive wearable devices or depth cameras (Plantard et al., 2015).

Other recent approaches include Consumed Endurance, which uses ergonomic models for the online assessment of mid-air planar interactive spaces (Hincapié-Ramos et al., 2014). Bachynskyi et al. (Bachynskyi et al., 2015) evaluated user muscle effort in 3D

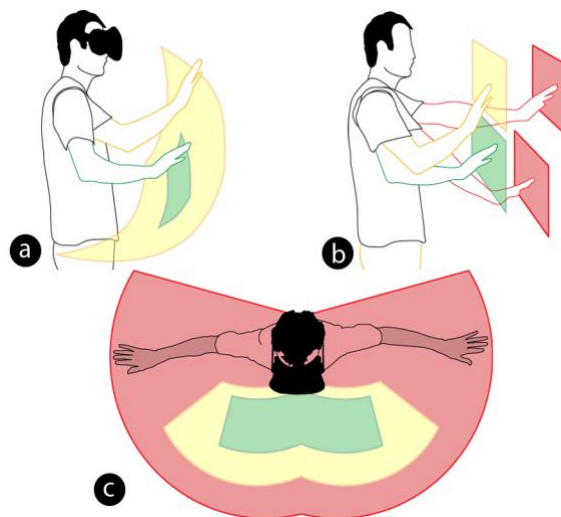
pointing tasks using EMG. They detect muscle activation and apply clustering techniques to identify movements with low muscle effort.

These techniques (illustrated in **Figure 4.6**), show a strong and consistent correlation between the space around the user and the most comfortable regions, with the middle area below the user’s chest being consistently ranked as most comfortable. These techniques also show how a change within the JND threshold allowed by visual dominance (i.e., a few centimetres) can have important effects on ergonomic scores. For instance, if our arm is fully extended aside with the hand at chest’s height, moving the hand just 5cm towards the belly, will reduce the RULA score from 5 (medium-high risk) to 2 (low risk).

ERG-O builds on these observations and methods, using these areas to guide our optimization methods and finding ergonomically acceptable mappings to retarget the interactive elements of the VE and improve comfort (**Figure 4.2**).

### 4.3 Erg-O: ergonomic Optimization for Redirected Interaction

ERG-O allows redirected interaction with the interactive elements of the VE. Leveraging visual dominance and ergonomic criteria, we reposition the physical location of the interactive elements (e.g., buttons on a cockpit), but we maintain their visual location. Besides reaching these elements from more ergonomic positions, users can still interact/reach any other point of the 3D space around them. Thus, ERG-O is the first manipulation technique to allow:



**Figure 4.2 RULA comfort mapping representation: Side view of the mapping highlighting in green the most comfortable zone to interact (a). Planes computed with *Consumed Endurance* (Hincapié-Ramos et al., 2014) approach (b), which are in agreement with our comfort mapping (c).**

- Open-ended, Multi-object retargeting (i.e., ERG-O can retarget several objects, with free hand movement and not knowing which object the user intends to reach).
- Isomorphic visual-to-physical mapping (i.e., only the visual points a user would be able to reach in reality are accessible. Each point of the visual space is mapped to one (and only one) point in physical space).
- Optimization-based computation of retargeting mapping (i.e., automatic computation of the physical location that leads to most ergonomic interaction, while minimizing visual-to-physical mismatch).

In order to realize these features, our approach is decomposed in two main stages. First, we create a *multi-object retargeting technique*. We partition the user's reachable space into tetrahedrons, with their vertices either on the boundary (limit of user's arm reach) or on a retargeted point. Each tetrahedron describes a volume in the visual space ( $\mathbf{V}$ ) and its matching volume in physical/retargeted space ( $\mathbf{P}$ ). However, their shapes will differ slightly, as a vertex on a retargeted point will have different coordinates in  $\mathbf{V}$  and  $\mathbf{P}$ .

This topology of matching tetrahedrons is key to Erg-O. When the (physical) hand is anywhere inside a physical tetrahedron, the virtual hand can be mapped to an equivalent point in the matching visual tetrahedron. When a physical hand reaches a vertex, the virtual hand is mapped to the equivalent vertex, whether this is a retargeted point (this allows our multi-object retargeting); or a boundary point (this still allows users to reach the extents of their natural interactive area). As the mapping only depends on the hand location, hands can be moved freely (open-ended).

At the second stage, we compute the *retargeting mapping*. This determines the best physical location to reach each visual element, using ergonomic and spatial criteria. Our technique dynamically adapts to the current interactive elements within user's reach (i.e., their number and position relative to the users will change as they move in the VE). We describe two example approaches to compute such *retargeting mappings*, one focused on maintaining the structural relationship between the interactive elements and a second one focused on improving ergonomic interaction.

These two stages are formally described in the following two subsections. For these explanations, we will make use of right-hand systems of reference, homogeneous coordinates (i.e., 3D points in A's coordinates as  $\mathbf{p}_A(x, y, z, 1) \in \mathbb{R}^4$ ) and homogeneous transformation matrices ( $\mathcal{M}_B^A \in \mathbb{R}^{4 \times 4}$ , to convert coordinates from A to B).

- 4.3.1 **Retargeted Manipulation: Bijective Mapping of Visual and Physical Spaces** Virtual hand interaction in VR usually assumes a direct correspondence between the physical space ( $\mathbf{P}$ ), around the user; and the virtual space ( $\mathbf{V}$ ), around their avatar. All points are mapped from one space to another directly through a transformation matrix (e.g.,  $\mathbf{p}_V = \mathcal{M}_V^P \cdot \mathbf{p}_P$ ).

Our multi-object retargeting requires a more complex mapping, at least for the points in space within user's reach. As introduced earlier, we use tetrahedrons as the basic space partitioning unit, and build two equivalent space partitioning trees (same structure), one for each space  $\mathbf{P}$  and  $\mathbf{V}$ . The steps required are detailed in the next subsections.

#### 4.3.1.1 *Tetrahedrons as Basic Space Partitioning Units:*

We first considered a sphere to represent the users' reachable space, however, since this geometry may lead to more complex computation and cost, we instead used a low-resolution approximation of a sphere i.e., a set of tetrahedrons grouped in an icosahedron layout, reducing thus the initial complexity and preserving low computational cost. Our approach uses a set of tetrahedrons pairs (see **Figure 4.3**), one defined in each space,  $\mathbf{P}$  and  $\mathbf{V}$ . Let  $\mathbf{T}_V = \{\mathbf{a}_V, \mathbf{b}_V, \mathbf{c}_V, \mathbf{d}_V\} \subseteq \mathbf{V}$  and  $\mathbf{T}_P = \{\mathbf{a}_P, \mathbf{b}_P, \mathbf{c}_P, \mathbf{d}_P\} \subseteq \mathbf{P}$  be a tetrahedron pair described by the visual coordinates and retargeted physical coordinates, relative to user's torso ( $\mathbf{T}$ ).

For each tetrahedron, it is possible to describe its own non-*orthogonal* and non-homogeneous system of reference, using their three edges and its first vertex as column vectors (matrix  $\mathcal{M}_V^T$  can be computed analogously):

$$\mathcal{M}_P^T = [(\mathbf{b}_P - \mathbf{a}_P)^+, (\mathbf{c}_P - \mathbf{a}_P)^+, (\mathbf{d}_P - \mathbf{a}_P)^+, \mathbf{a}_P^+] \quad (4.1)$$

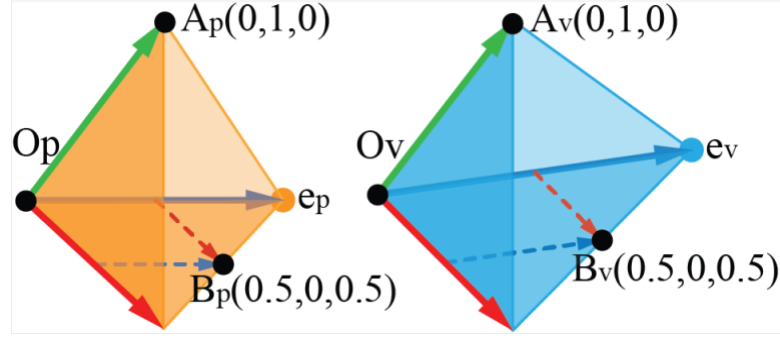
These matrices allow us to directly map any physical point  $\mathbf{p}_P$  inside  $\mathbf{T}_P$  to its analogous tetrahedron  $\mathbf{T}_V$ , by computing its local coordinates in  $\mathbf{T}_P$  and mapping the point to the same coordinates in the equivalent tetrahedron  $\mathbf{T}_V$ :

$$\mathbf{p}_V = \mathcal{M}_V^T \cdot (\mathcal{M}_P^T)^{-1} \cdot \mathbf{p}_P \quad (4.2)$$

By using this mapping strategy, the pair  $\{\mathbf{T}_P, \mathbf{T}_V\}$  now identifies two equivalent volumes in  $\mathbf{P}$  and  $\mathbf{V}$ , even if their shape is different (as in **Figure 4.3**). Thus, not only physical vertices  $\{\mathbf{a}_P, \mathbf{b}_P, \mathbf{c}_P, \mathbf{d}_P\}$  are mapped to their equivalent retargeted vertices  $\{\mathbf{a}_V, \mathbf{b}_V, \mathbf{c}_V, \mathbf{d}_V\}$ . Any other point inside  $\mathbf{T}_P$  can also be mapped to its equivalent in  $\mathbf{T}_V$  (e.g., a point on the edge  $\overline{\mathbf{b}\mathbf{c}_P}$  is mapped to point on the edge  $\overline{\mathbf{b}\mathbf{c}_V}$ ).

#### 4.3.1.2 Bounding the Interactive Space: Physical and Visual Trees

To build our space partitioning trees (we refer to them as tree **P** and tree **V**), we start by identifying the boundary of the interactive space around the user's torso in both spaces. We specifically approximate these as reduced icosahedrons, (with only 15 of the 20 tetrahedrons, as in **Figure 4.4(a-b)**). This geometry provides a basic structure, covering the space the user can reach with reduced complexity.



**Figure 4.3** Example of a tetrahedron pair, defining a volume in space **P** and its equivalent (slightly different) volume in **V**. Vertex **O** and edges are used to define their local systems of reference. This allows mappings any point in tetrahedron **P** to a single point in tetrahedron **V**, enabling retargeting.

Let  $T_P^i = \{t_P, p_{0P}, p_{1P}, p_{2P}\} \subseteq P$  and  $T_V^i = \{t_V, p_{0V}, p_{1V}, p_{2V}\} \subseteq V$ , with  $i \in [1,15] \subseteq \mathbb{N}$ , describe each of the 15 equivalent tetrahedrons in both spaces. Point  $t$  identifies the user's torso and the mapping between boundary points is computed as  $p_{jV} = \mathcal{M}_V^P \cdot p_{jP}$ ,  $j \in \{0,1,2\}$  (i.e., usual VR mapping described earlier). We use these 15 tetrahedrons to produce the two basic tree structures for **P** and **V**, with each tree containing 15 nodes in their first level and each tetrahedron node  $T_P^i$  in tree **P** having an analogous tetrahedron node  $T_V^i$  in tree **V** (shown in **Figure 4.1(c)**).

This tree structure is the seed for our multi-object retargeting mapping. Any point  $p_P$  around the user will be inside a unique leaf tetrahedron node  $T_P^i$  in tree **P**. Thus point  $p_P$  can be mapped to space **V** using  $T_V^i$ , as in Eq(4.2).

As each boundary vertex in tree **P** has simply been multiplied by  $\mathcal{M}_V^P$  to compute its matching vertex in **V**, the tetrahedron pairs have the same shape and our technique behaves like a traditional virtual hand (i.e., this is how the technique works if no interactive elements are within user's reach). Thus, current trees **P** and **V** act simply as an encapsulating boundary, allowing users to reach any point in **P** and **V**, but not the points beyond their natural reach.

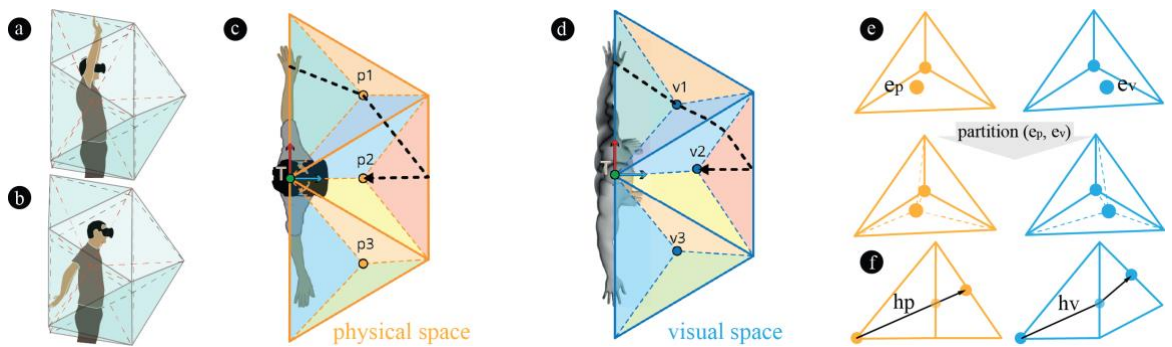
The following subsection will modify this initial behaviour, by adding the interactive (retargeted) elements to the basic tree structure. Each interactive element will add internal tetrahedron pairs, but their shapes will not match (see example in **Figure 4.4(c-d)**). Thus, the volume inside the basic encapsulating boundary will be distorted, to accommodate the retargeted interactive elements.

### 4.3.2 Retargeted Space Partitioning

We iteratively partition the basic tree described above (15 tetrahedrons pairs, vertices on the boundary), adding each of the interactive elements within user's reach. Let  $\mathcal{E}$  be the set of interactive elements. We model each element as a pair  $e = \{e_P, e_V\} \in \mathcal{E}$ , describing its coordinates in the visual and physical/retargeted spaces (the way we compute the pairs in set  $\mathcal{E}$  is explained in the next section).

For each point  $e_P$ , we determine the leaf tetrahedron node  $T_P$  it belongs to, subdivide it into four sub-tetrahedrons (as shown in **Figure 4.4(e)**), and add the corresponding nodes to tree structure  $\mathbf{P}$ . Each new sub-tetrahedron uses  $e_P$  as its first vertex (origin of coordinates, fourth column in Eq(4.1)), and 3 of the 4 vertices in  $T_P$ . The former leaf node  $T_P$  will keep a reference to  $e$  (and we will say " $T_P$  manages  $e$ ").

Next, for each sub-tetrahedron added to tree  $\mathbf{P}$ , we create its paired sub-tetrahedron  $T_V$  in tree  $\mathbf{V}$ . Thus, we use  $e_V$  as the first point and equivalent vertices in  $T_V$  (see **Figure 4.4(e)**), to ensure each sub-tetrahedron in tree  $\mathbf{P}$  remains equivalent to its paired sub-tetrahedron in tree  $\mathbf{V}$ .



**Figure 4.4** Summary of our manipulation technique: (a-b) Side 3D view of the boundary space enclosing user's interactive range. (c-d) Tetrahedron-based partitioning of the physical and visual space (simplified 2D view). Matching tetrahedrons highlighted on same colours. (e) An interactive element inside a tetrahedron will cause it to be subdivided in four tetrahedrons. (f) Continuity of interaction is assured when hand moves across tetrahedrons, but the direction and speed of motion can be affected.

This process produces the final equivalent tree structures for  $\mathbf{P}$  and  $\mathbf{V}$  shown in **Figure 4.4 (c-d)**. Tree  $\mathbf{P}$  maintains a hierarchical space partitioning structure. If a point is inside

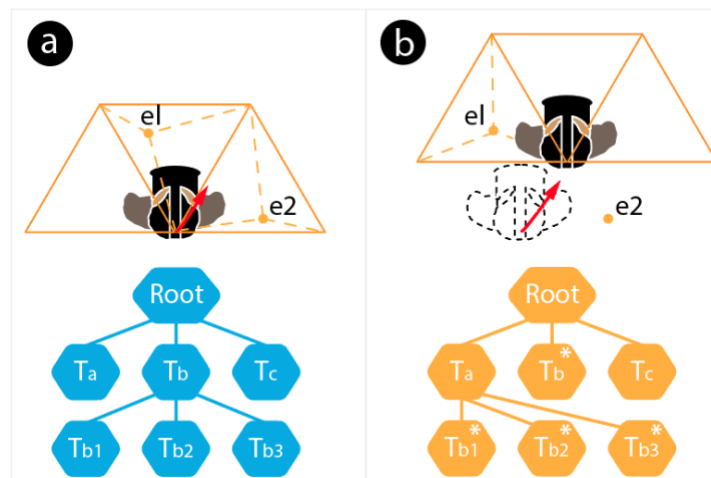
a tetrahedron node, it is also inside its parent's tetrahedron, but not inside any of the parent's siblings. This allows for efficient mapping of users' physical hand locations to retargeted visual locations, by finding the leaf node in tree **P** the hand is inside and mapping it to space **V** as in Eq. (4.2).

Also, neighbour tetrahedron nodes always share a common face (see **Figure 4.4(e)**). This ensures continuity in the mapping when the physical hand leaves a node in tree **P** and enters a neighbour. However, as the geometry of the equivalent tetrahedrons in **P** and **V** might differ, hand motion direction and speed can change (see **Figure 4.4(f)**). The effects redirections may have in an example hand trajectory are shown in **Figure 4.4(c-d)** (black path; redirections occur as the hand moves across tetrahedrons). However, our use of a minimum hierarchical topology of tetrahedrons (15 plus 4 per interactive element) minimizes the occurrence of these artefacts, and the fact that the hand is moving will reduce the chances of user's perceiving this change (Burns et al., 2005).

#### 4.3.3 Real-time Update of the Trees.

The position of the interactive elements relative to the user will vary as they travel through the VE. This will require changes to the tree structure, to maintain the hierarchical space partitioning properties of tree **P** (see **Figure 4.5**).

In each frame, we start by marking all tree nodes as dirty, and define an (initially) empty set  $\mathcal{E}'$ . We then proceed to iterate through the current set of interactive elements in  $\mathcal{E}$ . For a given  $e_p$ , managed by node  $T_p$ , if  $e_p$  is still inside  $T_p$ , this node and all of its children leaf nodes (i.e., not managing any other interactive element) are marked as clean. On the other hand, if  $e_p$  is inside a node other than  $T_p$ , we add  $e$  to  $\mathcal{E}'$  and remove it from  $\mathcal{E}$ .



**Figure 4.5** User displacements will change the mapping of interactive elements, causing the hierarchical tree structure to be recomputed.



At the end of this process all dirty nodes' children are removed from the trees (see  $e2$  in **Figure 4.5**). Set  $\mathcal{E}$  contains the elements which are still correctly located and are directly committed (i.e., update the location of tetrahedron's vertices). Finally, elements in  $\mathcal{E}'$  are re-introduced, using the iterative approach described in the previous sub-section.

#### 4.4 Computing Retargeting Mappings

The approach above describes a manipulation technique that can provide retargeted manipulation for a set of multiple interactive elements (given their physical and corresponding visual locations). The challenge still remains to find the optimum mapping  $\mathcal{E}$  for these points, based on ergonomics, spatial criteria and mismatch thresholds of visual dominance. Please note that only interactive elements (set  $\mathcal{E}$ ) are retargeted. Boundary points remain unaffected, to maintain the size of the user's reachable space.

As a first step to guide our retargeting approaches, we need to describe the metrics that will assess the quality of a retargeting mapping  $\mathcal{E}$ . We then report two example approaches to compute the retargeting mapping based on these metrics and different criteria.

##### 4.4.1 Quantifying Retargeted Mappings

Our algorithms will make use of three factors to evaluate the quality of the potential retargeting mappings. The final cost function for a mapping is computed as a weighted average, with the specific value of the weights depending on the retargeting approach used:

$$C(\mathcal{E}) = w_1 \cdot R_S(\mathcal{E}) + w_2 \cdot V_S(\mathcal{E}) + w_3 \cdot S_S(\mathcal{E}) \quad (4.3)$$

###### 4.4.1.1 Adapted Continuous RULA ( $R_S()$ ):

We use a metric inspired in the four first steps of the RULA process, as these are the steps providing an ergonomic score based on the position of the arms, which is the space our manipulation technique addresses.

Being initially an observational method, RULA uses broad ranges for the orientation of each joint, providing a discrete score for each range (e.g., a shoulder between +20°, is ranked as +1; 20-45°, is ranked as +2, etc.), with a final score for each arm between 1-9, associated to a risk level (neglectable, low, medium or high risks; associated areas for these scores are visible in **Figure 4.2** (a and c)).

This scheme allowed assessment of workers performing manual tasks through pictures or videos. However, when combined with our optimization methods, this results in a staircase function, with searches getting stuck in plateaus until the next step is reached (a change in RULA score). This then resulted in sudden changes in the retargeting mapping (e.g., when a user approached an interactive element, big changes in retargeting happened as it transitioned from one RULA score to the next one).

To prevent this and allow for smooth retargeting schemes, we simply take each angular range for each of the joints, and apply linear interpolation between the joint angle and the RULA scores for that range and joint. Then, for any given a point  $e_P$ , we used an IK algorithm (IKAN (Tolani et al., 2000)) to compute the angles of the three arm joints, keeping the angles providing most ergonomic (lower) score for disambiguation. The global score for a given mapping  $\mathcal{E}$  is then simply computed as  $R_S(\mathcal{E}) = \sum_{\{e_P, e_V\} \in \mathcal{E}} R_S(e_P)$ .

Please note this modified RULA score is used to compute mappings only. Our study used the usual RULA scores.

#### 4.4.1.2 Visual Dominance Mismatch Threshold ( $V_S(\mathcal{E})$ ):

In our study we explore the use of ERG-O for VR retargeted interaction, without making use of any type of tactile feedback. As such, the thresholds reported by Burns et al. (Burns et al., 2005) (up to 20cm) could be used. We however took a more conservative maximum mismatch of 10 cm, penalizing retargeting pairs where the distance between the visual and physical elements were likely to be detected. Thus, we defined our metric as  $V_S(\mathcal{E}) = \sum \|e_P - e_V\|, \forall \{e_P, e_V\} \in \mathcal{E} / \|e_P - e_V\| > 10cm$ .

#### 4.4.1.3 Spatial Relationship Preservation ( $S_S(\mathcal{E})$ ):

Our manipulation technique can map  $\mathbf{V}$  and  $\mathbf{P}$  spaces, based on any set of point pairs. This could result in mappings in  $\mathbf{P}$  space that hold not relation to the way elements are arranged in  $\mathbf{V}$  space. This metric penalizes mappings where the ratio of distances between physical pairs and visual pairs is not constant, as a way to preserve the topology between elements.

To do so, for each two points  $a, b \in \mathcal{E}$ , we measure their distance in  $\mathbf{P}$  and distance in  $\mathbf{V}$  and compute their ratio  $r(a, b) = \|a_P - b_P\| / \|a_V - b_V\|$ . To model that this ratio should be similar among all pairs (and penalize otherwise), we define  $S_S(\mathcal{E})$  as “the variance in  $r(a, b), \forall a, b \in \mathcal{E}$ ”.

#### 4.4.2 Optimization Methods to Compute Mappings

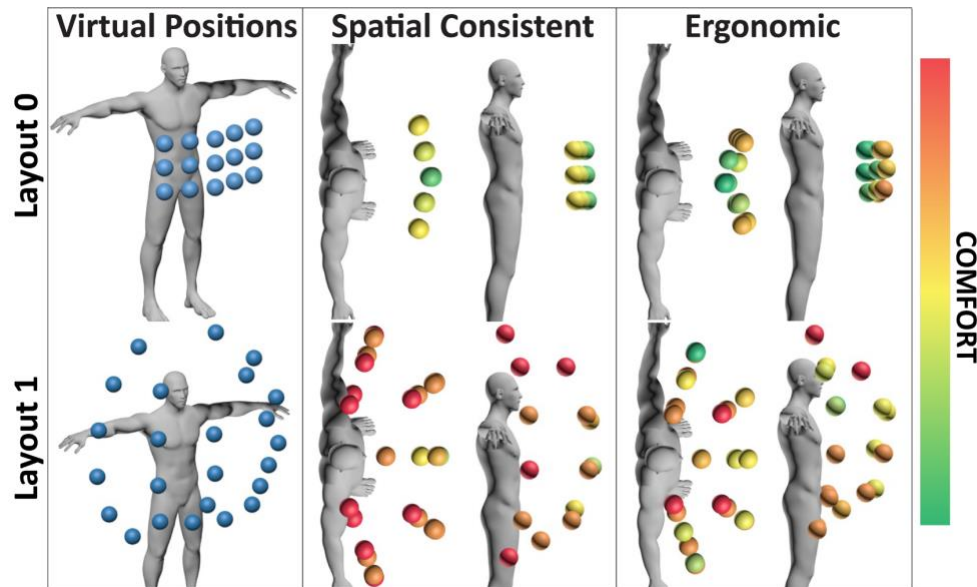
To illustrate our approach, we report two simple example approaches to compute retargeting mappings using the metrics described. The first approach is aimed at preserving the spatial structure between elements; while the second one loosens this criterion to reinforce ergonomics.

##### 4.4.2.1 Spatially Consistent Retargeting ( $S_R$ ):

This first approach is designed to improve ergonomics and maintain mismatch threshold but keeping the spatial relationships among the interactive elements intact (see second column in **Figure 4.6**). To do so, we use a scale transformation matrix  $S_P^V$ , centred on the user chest, to equally affect all interactive elements, with visual positions mapped to physical positions as  $p_P = S_P^V \cdot p_V$ .

This problem is modelled using a single variable  $k$ , to represent the scaling factor applied by  $S_P^V$ . Let  $t_V$  be the position of the user's torso and  $\{e_P, e_V\} \in \mathcal{E}$  the most distant element to  $t_V$ . If  $k \in (0, \|e_V - t_V\|)$ , all elements will stay within the reachable volume.

This technique uses a naïve linear search, testing 2000 potential  $k$  values within this interval to find the value that minimizes the cost function shown in Eq(4.3). Weights were empirically tuned to  $w_1 = 0.3$ ,  $w_2 = 0.7$ , with the last factor being ignored ( $w_3 = 0$ ), as this retargeting strategy inherently maintains spatial relationships.



**Figure 4.6 Comparison of the retargeting behaviour: Spatially Consistent and Ergonomic for two different layouts.**

This technique should not be mistaken for a simple motor-space scaling method. Only the interactive elements are retargeted, and the boundary still encloses all the reachable

space. Thus, redirections (**Figure 4.4(f)**) will still affect the hand (e.g., in our study, every time the user reaches from the belly to the target, or when reaching targets from the side).

#### 4.4.2.2 Ergonomic Retargeting ( $E\_R$ ):

This approach optimizes the position of each  $\{e_p, e_v\} \in \mathcal{E}$  independently. This loosens the constraints on spatial preservation from the previous approach, in order to achieve higher improvements for ergonomics.

This problem is modelled as a multivariable optimization approach, one for each of the XYZ coordinates of the  $e_p$  elements in  $\mathcal{E}$ . As in [Chapter 3](#), we use of Simulated Annealing (SA). The weights of the cost functions were empirically tuned to  $w_1 = 0.2$ ,  $w_2 = 0.4$  and  $w_3 = 0.4$ . Neighbours are computed by jittering a physical point  $e_p$  in the current mapping  $\mathcal{E}$  with a random direction and magnitude, for a maximum displacement of 5 cm (50% of our mismatch threshold). Transition acceptance between mappings follows the method by ([Kirkpatrick et al., 1983](#)). Let  $\mathcal{E}_1$  and  $\mathcal{E}_2$  be two potential mappings,  $C(\mathcal{E})$  be our cost function and  $T$  the current temperature. The probability of transitioning to from  $\mathcal{E}_1$  to  $\mathcal{E}_2$ . is computed as in Eq(4.4).

$$P(\mathcal{E}_1, \mathcal{E}_2, T) = \begin{cases} 1 & , C(\mathcal{E}_2) < C(\mathcal{E}_1) \\ e^{\frac{C(\mathcal{E}_1) - C(\mathcal{E}_2)}{T}} & , C(\mathcal{E}_2) \geq C(\mathcal{E}_1) \end{cases} \quad (4.4)$$

Our cooling schedule uses  $N_s=10$  step adjustments per temperature step,  $N_t=5$  temperatures steps per temperature change, cooling factor  $R_t=0.5$  and initial temperature  $T=180$ , testing nearly 18.000 possible retargeting mappings per optimization (frame). The results (best mapping) from a frame are used as the starting state for the next frame, as user displacements are likely to be small from frame to frame. This allowed us to produce satisfying results, while maintaining the real-time requirements of ERG-O, even with the relatively aggressive cooling schedule used.

#### 4.4.3 Analytical Comparison of Retargeting Approaches

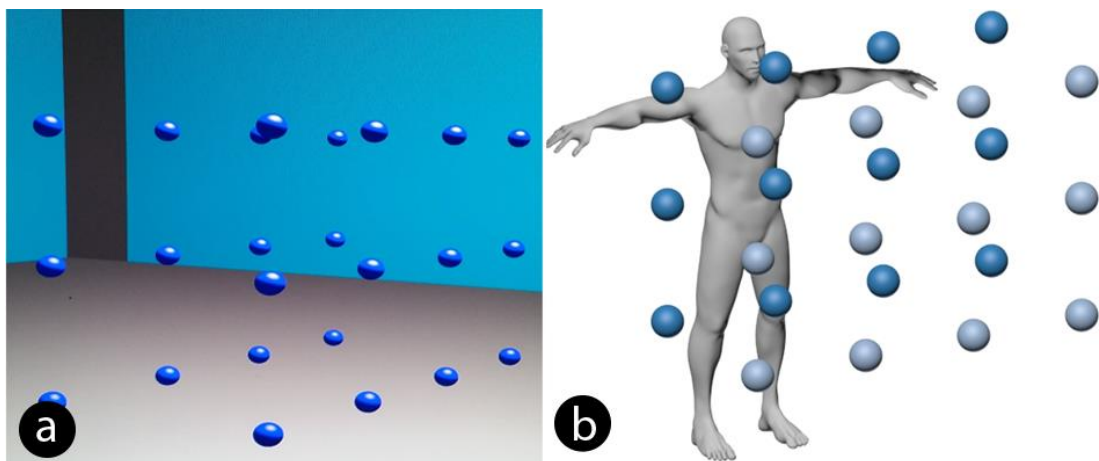
In this section, we analyse the differences in retargeting introduced by each of our example strategies. **Figure 4.6** shows two examples of interactive elements around the user. In the first example (top), the visual elements (in blue) are already at comfortable locations within user reach. In the second example (bottom), the interactive elements are evenly located around the user, close to the limits of their reachable space. Generally, these are uncomfortable positions, especially for lowest points and points above the user's chest.

As expected, *Spatially Consistent* retargeting ( $S_R$ ) repositions elements maintaining their spatial structure, while *Ergonomic Retargeting* ( $E_R$ ) affects structure, in order to enable more ergonomic interaction.

For the first example,  $S_R$  performed minor corrections (the displacement between visual and retargeted points within  $AVG + 6.8\text{cm}$ ,  $STDEV 6.1\text{cm}$ ), but still achieved improvements in ergonomics. The behaviour of  $E_R$  is more interesting. Although the retargeted distances were similar ( $AVG + 8.3\text{cm}$ ,  $STDEV 8.6\text{cm}$ ), the **Figure 4.6** (top, right) shows how  $E_R$  flipped the structure of the elements (i.e., from a concave to a convex shape). The resulting shape actually wraps around the central part of the ergonomic area (note the shape of RULA zones in **Figure 4.2(c)**), achieving much higher improvements.

This behaviour can be explained by looking at our definition of  $S_S(\mathcal{E})$ . By flipping the shape, the ratio of the distances between any two pair of points is actually preserved. This allowed ( $E_R$ ) to significantly improve ergonomic score for the nine central elements, although at the expense of decreasing the score of the elements at the edges. The wrapping to a curved shape shows  $E_R$ 's efforts to bring the edges back to more ergonomic locations, but metric  $S_S(\mathcal{E})$  did penalize this wrapping.

For the second example (second row in **Figure 4.6**),  $S_R$  shows a similar behaviour as in the first example ( $6.9 + 6.5\text{cm}$ ), while  $E_R$  presents a more aggressive behaviour ( $10.1+9.5\text{cm}$ ), especially the positions of low and high points (least comfortable). As expected, this results in higher ergonomic gains for  $E_C$ , but also higher spatial distortion.



**Figure 4.7 Experimental task: (a) Screenshot showing the selection task implemented in out testing environment and (b) third layout tested, with visual elements anchored to the world in two planes and forcing users to walk in order to reach them.**

## 4.5 User Study

The previous sections motivate the need for Erg-O and provide a formal definition for the technique. The current section will evaluate the usability of the technique in a VR selection task. We compare our two examples of retargeting strategies against a traditional virtual hand technique, to gain insight on the improvements obtained for ergonomic interaction and the influence that distortions (due to retargeting and space warping) could have on interaction.

### 4.5.1 Task and Environment

We implemented a target selection task to test our technique, with 30 trials per task. At the beginning of each trial, users could see a range of blue spheres (i.e., interactive elements) floating in the space in front of them. Two spheres were highlighted in green (instead of blue), to inform users of the targets of their selection, and a regressive countdown from 5 to 0 was shown. When the countdown finished, users touched their belly to select the hand they wanted to use for that trial, and proceeded to touch the highlighted spheres. An auditory cue notified users when they had correctly selected each target. The pair of spheres to select was randomly chosen, but both spheres were reachable with a single hand.

The environment was implemented using C++ and OpenGL. We used an Oculus Rift DK2 for display, OptiTrack to achieve a larger tracking volume and Kinect v2 for skeletal tracking. Projection matrices and barrel distortion meshes were replicated from Oculus SDK v1.7. Conventional speakers were used for audio feedback.

### 4.5.2 Layouts Tested:

We tested the techniques using three different layouts, to assess their performance under several usage scenarios. In some layouts (L1 and L2) we wanted spheres to stay in specific areas relative to the user (e.g., in comfortable/ uncomfortable points). In these cases, the grid of spheres was anchored to the users' lower torso, so that they would stay at these fixed areas even if the users moved. Upper torso (i.e., chest) was avoided as an anchor, as its orientation can change when users reach towards an object due to accompanying movement of the shoulder.

During tests, we measured users' arm span  $A$ . All distances and positions describing our layouts are relative to  $A$ , but in our explanations, we will report the equivalent value in centimetres for a reference user with  $A=170$  cm.

#### 4.5.2.1 Ergonomic Layout (L1)

This layout (shown earlier in **Figure 4.6**) consisted of 15 spheres, placed in a 5x3 grid in front of the user's lower torso and at a distance of  $0.21 \cdot A$  (~36 cm). This is an agreed zone for comfortable interaction (e.g., middle ground between the comfortable plane used in (Hincapié-Ramos et al., 2014) and the volume receiving a RULA score of 1). This was chosen as a worst-case scenario to test against our technique, because: a) there is little room for improvement due to ergonomic retargeting (spheres are already at comfortable locations); and b) users will still suffer from the distortions and loss of linearity (**Figure 4.4(f)**) introduced by our retargeting strategies.

#### 4.5.2.2 Limits of Reach Layout (L2)

This second layout (also displayed in **Figure 4.6**) consisted of 24 spheres, evenly distributed along the limits of users' reachable space at  $0.44A$  (~75 cm) and anchored to the users' lower torso, as above. In contrast to L1, this layout should provide best ergonomic improvements, but at the same time, it will introduce more aggressive retargetings. This can increase spatial distortion, which could hinder motor control and affect the selection task. Thus, L2 should help illustrate the extent of the benefits of Erg-O for ergonomics and the impact of its redirections.

#### 4.5.2.3 World Fixed Layout (L3)

This last layout (shown in **Figure 4.7(b)**) is based on a more generic scenario where elements are fixed in the VE (instead of anchored to the user) and distributed across a bigger volume, forcing users to walk to them in order to interact with them. More specifically, 24 spheres were evenly distributed over two vertical planes, spanning across  $1.4A \times 0.8A$  (238 cm x 136 cm). Lowest and highest spheres were placed at heights  $0.4A$  (68 cm) and  $1.2A$  (204 cm) from the floor, forcing users to reach both low points and points above their heads. Both planes were separated by a distance of  $0.5A$  (85 cm), ensuring elements in one plane would not be in reach from the other plane. Besides testing a more generic scenario, this layout allowed us to see the influence of a varying retargeted mapping (i.e., the retargeting for each sphere changed as user moved, as this changed the sphere's position relative to the user).

### 4.5.3 Experimental Design

In the experiment, we compared three techniques: Natural virtual hand (*N*), *Spatially Consistent* retargeting (*S\_R*) and Ergonomic retargeting (*E\_R*). We adopted a 3x3 full

factorial design, with factors being the technique ( $N$ ,  $S\_R$  or  $E\_R$ ) and layout ( $L1$ ,  $L2$  or  $L3$ ), counterbalanced following a Latin Square design.

The experiment was conducted with 12 participants (10 male, and 2 females between the ages of 21 and 35). We collected 3240 trials (12 participants, 9 blocks, 30 trials each). Each participant was tested individually and the experiment took approximately 45 minutes per participant.

#### 4.5.3.1 *Dependent Variables Measured*

Participants were asked to fill a user comfort and physical effort questionnaires after each block (technique).

The experimental software recorded: trial completion time (TCT), and the length of the real (RP) and virtual hand paths (VP) for each trial. TCT measured the time between the user touching the first and second highlighted spheres. Path lengths (RP and VP) were measured as the ratio between the length of the path followed by the (real or virtual) hand, divided by the linear distance between the spheres (Zhai et al., 1998). This allowed comparisons across paths of different lengths and measured effectiveness (deviation from optimum) for the interaction. Conventional RULA scores were also recorded when users selected each sphere, to test if our mappings actually improved ergonomics.

## 4.6 Results and Analysis

For all analysis presented in this section a repeated measures ANOVA was conducted to compare the effect of the 3 techniques (Natural ( $N$ ), Spatially Consistent ( $S\_R$ ) and Ergonomic ( $E\_R$ )) on mean time, effort and path length. Outliers were filtered out (i.e., mean  $\pm$  2 standard deviation), removing 284 trials (2.83% of samples). Post-hoc comparisons used Bonferroni corrections for each case.

We start the analysis by looking at the general behaviour of each retargeting approach, and then focus our analysis on each of the different layouts tested, to get further insight on how the approaches behave in different scenarios.

### 4.6.1 General Analysis of Retargeting Approaches

#### 4.6.1.1 *Retargeting Approach vs Time*

Results showed a significant effect on the average time required to complete the selection task ( $F(2,4)=279.67$ ,  $p<0.001$ ), depending of the type of retargeting. Post-hoc comparisons using Bonferroni corrections showed significant differences. Specifically,  $S\_R$  ( $M=0.945s$ ,  $SD=0.473s$ ) was faster TCT than  $E\_R$  ( $M= 1.043$ ,  $SD= 0.473s$ ),  $p=$



0.001; and also, faster than *Natural* ( $M= 1.03s$ ,  $SD= 0.734s$ ),  $p= 0.005$ . No such differences were found between *E\_R* and *Natural*,  $p= 1$ . These findings suggest that *E\_R* and *Natural* conditions behave in a very similar way, but the use of the *Spatially Consistent* retargeting approach can lead to lower task execution times.

#### 4.6.1.2 Retargeting Approach vs Comfort (using RULA)

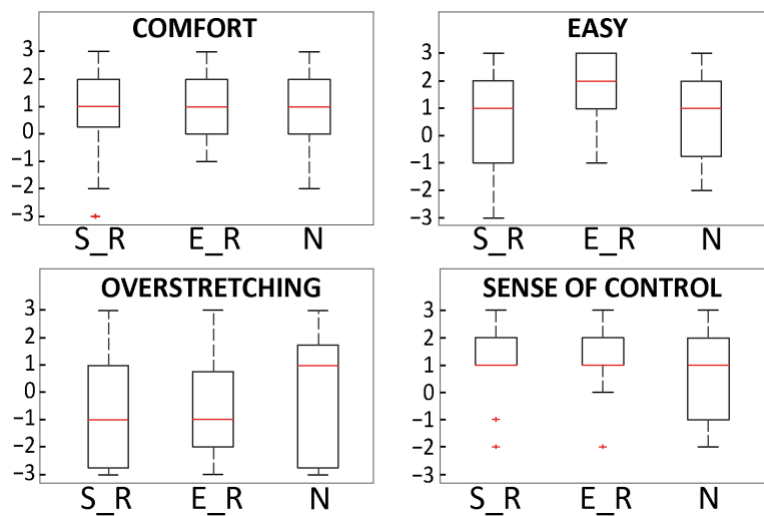
Comfort was defined using the RULA scores calculated. Results showed a significant effect on comfort  $F(2,4)=26.06$ ,  $p<0.001$  depending on the retargeting approach used. Post-hoc tests with Bonferroni corrections show significant difference ( $p=0.002$ ) between *E\_R* ( $M=1.6078$ ,  $SD=1.0934$ ) and *Natural* ( $M=1.7767$ ,  $SD=1.1175$ ). No other differences in comfort were found. This suggests that the *Ergonomic (E\_R)* retargeting approach can reduce user effort, possibly enabling longer usage periods before the user gets tired.

#### 4.6.1.3 Retargeting Approach vs Length

This analysis showed no effects of retargeting type on path length performance. This could indicate that the thresholds used to optimize the target positions ( $\sim 10cm$ ) were too small compared to the hand displacements required to complete the task, as to represent a significant difference.

#### 4.6.1.4 Retargeting Approach vs Self Reports

**Figure 4.8** shows the results of the questionnaires filled after each retargeting condition block (*S\_R*, *E\_R* and *Natural*). We used a Likert scale from -3 to 3 to assess the comfort, easiness of reachability, sense of control and overstretching. The boxes represent the interquartile ranges (IQRs) and the whiskers represent the confidence interval.



**Figure 4.8** Box plots for the results of our questionnaires: Horizontal red bars represent medians, and boxes represent the interquartile ranges (IQRs). Whiskers stretch to the data points that are within the median  $\pm 1.5$  IQR.

Participants perceived all techniques as similar in terms of comfort, which indicates that the blocks were not long enough as to make the improvements to ergonomics (RULA) become a driving factor for effort in this task.

Approach  $E\_R$  was perceived as the approach allowing easier reachability and less overstretching. More interesting, users reported that they felt more control when using any of the retargeted conditions ( $S\_R$  or  $E\_R$ ) than using  $N$ .

This could be a result of both  $S\_R$  and  $E\_R$  placing the objects at slightly closer physical positions of each other. This would cause the index of difficulty of the task (Fitts's law) to decrease, making them easier to reach and potentially affecting the users' assessment about their sense of control. However, as no significant differences were found for path length (hand motion not significantly shorter than in  $N$ ), we cannot strongly support this hypothesis.

In any case, the fact that  $S\_R$  or  $E\_R$  did not actually receive worse scores for control was found a very positive result. This seems to indicate that the spatial distortion and the artefacts related to linearity (**Figure 4.3(f)**) still allow fluent interaction, confirming our empirical observations.

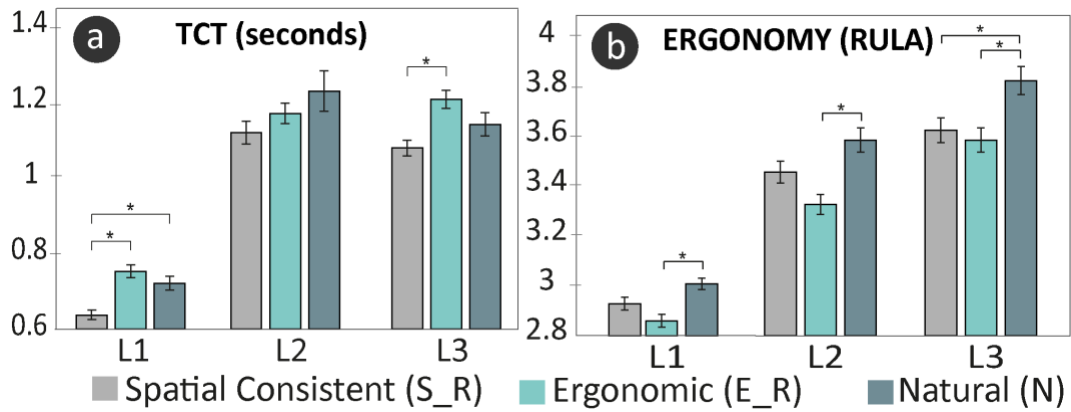
#### 4.6.1.5 Retargeting Approach vs Layout

##### *Layout 1*

The results showed a significant effect of retargeting on TCT ( $F(2,4)=12.295$ ,  $p<0.001$ ), for Layout 1. Post-hoc comparisons showed significant differences in time, with  $S\_R$  ( $M=0.640s$ ,  $SD=0.234s$ ) leading to lower TCT than either  $E\_R$  ( $M=0.749s$ ,  $SD=0.316s$ ),  $p<0.001$ ; or *Natural* conditions ( $M=0.722s$ ,  $SD=0.348s$ ),  $p=0.001$ .

Retargeting approach also influenced Comfort ( $F(2,4)=8.92$ ,  $p<0.001$ ). Pairwise comparisons showed significant differences, with  $E\_R$  ( $M=2.86$ ,  $SD=0.502$ ) providing better scores than  $N$  ( $M=3.01$ ,  $SD=0.529$ ),  $p<0.001$ , even if objects in Layout 1 were already at comfortable positions.

These results confirm general expectations about  $S\_R$  and  $E\_R$ , with the first one improving performance, while the second one improved comfort. However,  $E\_R$  not being faster than  $N$  could indicate  $E\_R$  was not good for performance. The objects being at more comfortable locations, and the arguably smaller index of difficulty of the task, should both benefit  $E\_C$  for TCT. We believe the more aggressive redirections could make users rely more on visual feedback, making them perform more slowly.



**Figure 4.9** Average measurements per strategy and layout: (a) Time completion task and (b) RULA scores during the task. Significant difference between retargeting strategies for each layout are represented by ‘\*’.

### Layout 2

Our results showed no significant effect of approach on TCT ( $F(2,4)=2.664$ ,  $p=0.07$ ) for Layout 2. Retargeting a few centimeters was probably not a significant advantage for the longer hand displacements required in this task.

The analysis however showed significant effects on Comfort ( $F(2,4) = 8.238$ ,  $p<0.001$ ), with paired analysis indicating that *E\_R* ( $M=3.33$ ,  $SD=0.75$ ) led to better comfort than *Natural* ( $M=3.59$ ,  $SD=0.932$ ),  $p<0.001$ , and confirming the general trend of these techniques, also for scenarios involving large arm movements.

### Layout 3

As shown in **Figure 4.9**, retargeting strategy showed significant effects for both TCT ( $F(2,4)=5.628$ ,  $p=0.004$ ) and Comfort ( $F(2,1077)= 5.902$ ,  $p=0.003$ ) in this layout. Post-hoc comparisons showed participants were faster using *S\_R* ( $M=1.08s$ ,  $SD= 0.385s$ ) than with *E\_R* ( $M= 1.2s$ ,  $SD= 0.437s$ ),  $p= 0.002$ , but no significance was found compared to *N* ( $p= 0.282$ ).

In terms of comfort (see **Figure 4.9(b)**), the scores were generally high (uncomfortable). *Natural* ( $M=3.83$ ,  $SD=1.076$ ) led to worse results than either *E\_R* ( $M=3.59$ ,  $SD=0.924$ ),  $p=0.005$ , or *S\_R* ( $M=3.63$ ,  $SD=0.950$ ),  $p=0.02$ . We observed users tended to walk towards the targets with their arms fully extended (i.e., focusing on performance, rather than interacting comfortably), which probably blurred the differences between techniques. However, instructing participants to interact comfortably (rather than quickly) could have produced a similar bias (i.e., walk until the target is in comfortable reach and then select; this would also result in no differences in comfort across techniques). Instructing them to

walk to specific points before selecting, would have implicitly fixed the location of targets relative to the user, blurring differences with L1 and L2.

#### 4.7 Discussion

The approach described (Erg-O) uses controlled warping of the visual and physical space around the user, to enable multi-object retargeting in an open-ended fashion. Our two example retargeting functions ( $E\_R$  and  $S\_R$ ) also helped us illustrate how Erg-O can be used to improve ergonomic interaction in VR. In spite of warping space (i.e., virtual hand not following the exact motion of the real hand), both strategies improved ergonomic scores without decreasing performance (actually,  $S\_R$  resulted in better TCT than N for some scenarios); sense of control or complexity.

Our example strategies also highlight the importance of the retargeting function. First, they can result in very different behaviours ( $S\_R$  being generally better for TCT, while  $E\_R$  improved ergonomics). Using other metrics for ergonomic assessment (e.g., Jack (Badler et al., 1993) or (Hincapié-Ramos et al., 2014)), other functions or weight distributions could produce different results. Second, even simple functions, such as  $S\_R$ , can produce good results.

The importance of spatial preservation was also highlighted (see artefacts produced by  $E\_R$ , in **Figure 4.6**). For example, consider two elements, with A visually to the left of B. A function retargeting B to the left of A would result in undesirable discontinuities for interaction. Similarly, tetrahedron pairs with very different shapes or volumes would result in strong redirections and significant changes in speed, and the retargeting function should avoid this. Beyond the examples presented in this chapter (multi-object retargeting to improve ergonomics in VR), varying specific aspects of Erg-O can adapt it to other application scenarios.

For instance, boundaries **P** and **V** were kept equal and always anchored to the user's and avatar's chest. Scaling the boundary in **V**, would enable interaction with distant objects in the VE (i.e., similar to Go-Go, but allowing multi-object retargeting inside). Detaching tree **V** from the user's chest and moving it to a distant point of the VE could replicate the HOMER (Bowman et al., 1997) technique.

Also, unlike in the example presented in the chapter, the shape of the boundaries **P** and **V** do not need to match. This could be useful for users with limited limb mobility, as the boundaries of tree **P** can be tailored to circumscribe the physical space the patient can

reach. Tree  $\mathbf{V}$  could still circumscribe the reachable space for a person with normal mobility, and our isomorphic mapping would allow patients with reduced mobility to interact within all this space.

The internal topology of the tree (retargeted points and resulting tetrahedrons) could be used to further refine this mapping. In the case of users with limited mobility this could be used to avoid uncomfortable poses or to provide adequate levels of resolution to specific parts of the space, based on the patient's motor skills and condition. As a particular example, this could apply to children with mental palsy or spasticity, to create novel range of motion exercises (Dunne et al., 2010) or building games (e.g., LEGO) exploiting the spatial properties of VR to improve cognitive skills (Cheng et al., 2014).

Taking the opposite approach, the retargeting strategy could be tailored to force specific poses in the patient (e.g., most of the visual space mapped to higher locations in the user's physical space, forcing the user to lift his arms). This could be applied for rehabilitation or physical training purposes.

Our solution could also be applied to surgical simulations, such as (Spillmann et al., 2013), in which retargeting is currently limited to two dimensional surfaces.

## 4.8 Conclusion

In this chapter, we presented Erg-O, a multi-object retargeting technique for manipulation in VR. The visual location of one or more interactive objects (e.g., buttons) is maintained, but users can reach them from more ergonomic locations. Users can move their hands freely, and they can also reach any other points (not only retargeted elements). We achieve this by creating a mapping between the visual and physical space that warps the user's reachable space according to the location of the retargeted elements.

We presented a formalization of our manipulation technique, and also described two example retargeting strategies to compute the best physical retargeted positions for interactive elements, according to spatial and ergonomic criteria. We finally evaluated the performance of these example retargeting techniques compared to a traditional virtual hand (baseline). Results from our study demonstrated the potential of our technique to improve ergonomics, without significant effects on performance or sense of control. We finished the chapter by discussing relevant aspects related to the use of Erg-O in other scenarios, as well as identifying other possibilities and application scenarios where Erg-O can be applied.

## Chapter 5

# NaviFields: Relevance fields for adaptive VR navigation

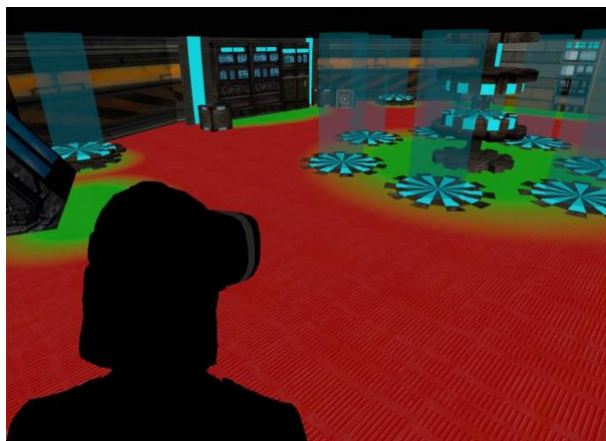
Similar to the previous chapter, in this chapter we explore the second stage of interaction (i.e., ITes). However, here we refer to a different BITa according to (Molina-Masso et al., 2008). We focused on *navigation*, another primary interaction task (Bowman et al., 1999). We led our exploration around the problem of limited real space for VR *navigation*, giving particular attention to “manoeuvring” (i.e., the stage after a relevant target has been reached) (Bowman et al., 2001), where precise exploration is involved, focusing on a seamless transition between *navigation* and *manipulation* tasks in VR. Our approach in this chapter aims to better exploit the human potential for navigation (i.e., enabling vestibular cues through actual physical displacement) while addressing physical space constraints. We present a *scale-adaptive* navigation technique that addresses this limitation by applying computational methods into an *allocentric* metaphor (i.e., considering the space relevance rather than the user viewpoint) that uses non-isomorphic mappings to extended VR navigation while maintaining naturalness in relevant zones for precise interaction (1:1 movement ratio).

## 5.1 Introduction

Physical displacement in Virtual Reality (VR), where the viewpoint is directly controlled by the user’s head motion (Bowman et al., 1997), stands as the most natural navigation techniques for VR and benefits both interaction and sense of presence (Bystrom et al., 1999). However, limitations in tracking technologies (i.e., reduced tracking volume) or in the actual physical space available (e.g., empty space in a user’s living room) practically constraint the size of the Virtual Environment (VE) that users can navigate.

Many techniques have been proposed to overcome these limitations, allowing users to navigate a virtual space bigger than the actual physical space available. Treadmills can achieve this while maintaining a natural navigation metaphor, but expenses and the need to deploy a (potentially bulky) hardware element in the user home can limit their adoption. Teleportation or Steering (e.g., using head or hand orientation to control the direction of motion) techniques enable unconstrained navigation. However, they also break the metaphor of using physical displacements to move in VR, which can affect user's presence (Bystrom et al., 1999), and might only be well suited for specific scenarios (e.g., while teleportation could be adequate in a sci-fi action game in VR, it might be ill suited for simulation or training scenarios). In this chapter, we explored not only more natural navigation in VR through physical locomotion, but also explored object interaction during the exploration. For that reason, hand-guided navigation is not employed, as it does not mimic the way we navigate in the real world.

In this chapter we propose *NaviFields*, a VR navigation technique that maintains the physical displacement metaphor (Bowman et al., 2001), while expanding the size of the VE that users can navigate. With our technique users' head displacement is scaled according to their position in the VE. If the user is in an interactive area requiring fine control (e.g., assembling machinery parts), the viewpoint will follow the real motion of the head (1:1 direct navigation). In contrast, while travelling across connecting spaces (e.g., the corridor leading to the maintenance room), motion will be gradually scaled, requiring smaller displacements to cover bigger distances.



**Figure 5.1** NaviFields approach scales user displacements dynamically, enabling natural 1:1 navigation inside areas where precise interaction is required (green) while progressively increasing their movements outside these areas (yellow, red). This allows larger navigable spaces, maintains the natural displacement metaphor and allows precise manoeuvring where necessary.

We do this by creating a *navigation field* (see **Figure 5.1**) describing the relevance (i.e., requirements for precise motion) of each point of the VE, and then use this field to determine the scaling applied to user displacements at each point. This practically compresses the overall size of the VE, but retains a direct 1:1 navigation within highly interactive areas. This is useful in, for instance, a training VE for a factory, where there is a well identified set of stations for machinery control (interactive areas requiring fine control), but at the same time the user needs to build a mental model of how these stations are arranged in the real factory.

This chapter contributes a formal description of the *NaviFields* technique. We then report a quantitative testbed evaluation, assessing low-level travel tasks (search travel and manoeuvring ([Bowman et al., 2001](#))), and explore the effect of using varying scaling factors, travel path lengths, manoeuvring complexity and user poses. We identify the potential and limitations of *NaviFields*, by comparing its performance to the use of: **a)** physical displacements with a constant scaling factor (naïve alternative technique that maintains physical displacement and covers bigger navigable spaces); and **b)** natural navigation (baseline comparison, best case scenario).

Our results show *NaviFields* can be comparable to natural navigation in manoeuvring tasks, and still allows good performance for search travel tasks. When compared to the homogeneous scaling technique, *NaviFields* showed better performance for all factors assessed (travel, manoeuvring and user preference). Our results also provide useful insight about the effects of scaling or user pose in travel and manoeuvring tasks; or the effects of *Drift* in navigation techniques based on differential tracking, applicable to other non-isomorphic techniques. We finish the chapter discussing the opportunities and application scenarios *NaviFields* enable.

## 5.2 Related Work

We review prior work in two related areas: (1) VR navigation techniques; and (2) Dynamic control of the viewpoint.

### 5.2.1 VR Navigation Techniques

Navigation is identified as a fundamental task for VEs, being usually decomposed ([Bowman et al., 2001](#)) into wayfinding (the cognitive process related to navigation) and travel (the actions executed to reach a destination). Travel can be further divided into:



exploration (roaming with no explicit goal), search (there is a specific goal), and manoeuvring (small displacements, precise control of the viewpoint required).

Natural walking stands as the most natural and effective navigation technique ([Usoh et al., 1999](#); [Riecke et al., 2010](#)), not involving additional controls and leveraging our oculomotor control and vestibular systems, with positive effects to understand the environment ([Cutting, 1997](#)). However, this method of locomotion is only feasible when the 3D world is (at most) as big as the working volume of the tracking system. Hybrid approaches complement walking with other techniques, such as joysticks to travel in specific directions ([Cirio et al., 2009](#)), or controllers to teleport to other locations (e.g., commonly used in HTC Vive games). However, joysticks create the feeling of flying rather than walking ([Lécuyer et al., 2006](#)), and teleportation hinders navigation skills ([Darken et al., 1993](#)).

Redirected walking techniques exploit change blindness ([Peck et al., 2009](#)), giving the illusion of naturally walking a large VE while keeping users within the tracking volume. Techniques proposed make use of rotational gains ([Razzaque et al., 2001](#)), translation gain ([Williams et al., 2006](#)), space substitution ([Suma et al., 2011](#)) or distractors ([Peck et al., 2010](#)). However, the tracking spaces needed are still large (e.g., 6.5x6.5m in ([Peck et al., 2009](#))).

Walking in place (WIP) techniques also involve physical displacement (i.e., navigation controlled by the movements of the user's body), but with no actual translation of the user. Thus, users simulate walking, and the movement of their feet ([Slater et al., 1995](#)), heels ([Feasel et al., 2008](#)) or knees ([Wendt et al., 2010](#)) is used to control translation in the VE. These techniques have also been adapted for mobile VR devices ([Tregillus et al., 2016](#)) and seated environments ([Terziman et al., 2010](#)). Although regarded as immersive and natural ([Slater et al., 1995](#)), WIP techniques do not provide the same vestibular cues than walking (e.g., no inertia). Delays in detecting the start/stop of the motion can also affect presence ([Templeman et al., 1999](#)) and encumber manoeuvring ([Feasel et al., 2008](#)).

Hardware solutions using linear ([Stanney, 2003](#)) or omnidirectional treadmills ([Darken et al., 1997](#)), or with the user walking inside a rotating sphere ([Fernandes et al., 2003](#); [Medina et al., 2008](#)), provide a closer match to natural locomotion. However, they still do not produce the same proprioceptive perception as real walking ([Bowman et al., 2001](#)).

The need to deploy a bulky and expensive hardware element in the users' home can also limit their adoption.

Steering techniques loosen the role of body motion. They use the direction of the user's head (Fuhrmann et al., 1998), torso (Laurel et al., 1994) or hands (Bowman et al., 1997) to determine direction of motion, but require additional control commands (e.g., joysticks, gestures) to trigger motion or determine speed. Solutions using joysticks, gamepads or mouse and keyboard have also been extensively user (e.g., games), but they negatively influence spatial orientation (Lathrop et al., 2002) and the sense of presence (Usoh et al., 1999).

### 5.2.2 Dynamic Control of the Viewpoint

Dynamically scaling the translation speed of the viewpoint has been mostly applied for less immersive navigation techniques (i.e., not involving physical displacement).

Mackinlay et al. (Mackinlay et al., 1990) developed one of the pioneering techniques, with some similarity to the *NaviFields* technique. When a user selects a target destination, a logarithmic function allows fast displacements along big distances, progressively slowing down as the user reaches the destination. Argelaguet's approach allows open-ended navigation (not based on selection of a target destination), but uses the optical flow in the user's view to adjust the navigation speed based on the user's perception of motion (Argelaguet, 2014). Lecuyer et al. use a model of the head's lateral motion, rotation and eye fixation, affecting viewpoint motion to improve the sensation of passive walking in the VE (Cirio et al., 2009). In these techniques, motion speed is not connected to the meaning/relevance of the objects in the environment. Freitag et al. adjust travel speed based on viewpoint quality (i.e., how informative a viewpoint is), sharing some conceptual similarity with our approach (Freitag et al., 2016).

The dynamic modification of viewpoint scale and speed is much more uncommon for VR techniques involving physical displacement, as these are mostly focused on providing a 1:1 mapping between the user's real and virtual displacements.

Redirected walking techniques have made use of subtle changes in scaling, to alter users' paths without translation gains becoming noticeable (Steinicke et al., 2010). Multi-scale techniques scale the size of the user (rather than its speed), to interact with the VE at microscopic or macroscopic levels (Kopper et al., 2006).

Use of higher scaling factors to navigate bigger spaces have been even less common. Williams et al. scale user's physical displacement using a constant scaling factor (Williams et al., 2006). Interrante et al. couple the scaling factor, to the speed of the user's real head (Interrante et al., 2007). While allowing close to natural navigation at low speeds, the viewport will move exponentially faster, the faster the user moves in reality. LaViola et al. (LaViola Jr et al., 2001) use their feet to interact with a World-In-Miniature (WIM) (Bowman et al., 2001). Little displacements on the WIM cause large displacements of the user, and scale can also be adjusted with foot gestures. In all these cases, viewport speed is controlled by users' actions alone (egocentric), and not by the contents of the VE (*allocentric*, as in *NaviFields*).

The closest match to the proposed *NaviFields* technique can be found in (Song et al., 1993). This technique identifies a sphere in the VE, describing the primary space for interaction. Natural navigation is available inside the inner sphere, but motion is scaled exponentially outside this sphere. Like *NaviFields*, this allows for bigger navigable spaces, using physical displacements and with scaling being driven by the structure of the VE (i.e., *allocentric*, instead of egocentric). It can thus be considered as a particular instance of the fields our technique covers, but it does not deal with the interactions among several areas and cannot address the relevance of each point of the VE individually.

### 5.3 NaviFields: Adaptive VR Navigation

*NaviFields* uses the known location of the interactive areas within the VE, enabling natural 1:1 navigation within those areas, while gradually speeding up displacements when travelling between interactive areas. In practice, this increases the navigable space, retains a physical displacement metaphor in all the VE and 1:1 natural displacement in places demanding precise navigation (manoeuvring) or interactive tasks (precise manipulation).

The following sections describe the mathematical modelling of the adaptive navigation and the description of the *navigation field* (that determines the scaling applied at each point of the VE). For our explanations, systems of reference will be noted as capital letters, with **U** referring to the user's head system of reference; **T** referring to the system of reference of the tracking system; and **W** referring to the system of reference of the virtual world. We will make use of right-hand systems of reference, homogeneous coordinates (i.e., points in **A**'s coordinates as  $\mathbf{P}_A(\mathbf{x}, \mathbf{y}, \mathbf{z}, 1) \in \mathbb{R}^4$ ) and 4x4 matrices

( $\mathbf{M}_B^A \in \mathbb{R}^{4 \times 4}$ , to convert coordinates from A to B). This notation will aid reproducibility and ease explanation of our technique in comparison with homogeneous scaling.

### 5.3.1 Modelling Navi-Fields: Differential Tracking

Physical displacement techniques usually rely on a bijective mapping between the real space (tracking volume) and the navigable space in the VE. In other words, each point in the real world is uniquely mapped to a point in the VE and vice-versa. Our technique breaks this bijective mapping in order to dynamically scale displacements according to the location of the user (inside an interactive area or a transition area).

This can be illustrated comparing *NaviFields* to the use of a homogeneous scaling factor (Williams et al., 2006) (shown in Eq(5.1)). In this technique, the position of the user's head at any specific point in time  $\mathbf{M}_T^U(t)$  is scaled by a constant scale matrix  $\mathbf{S}(\mathbf{k}, \mathbf{1}, \mathbf{k})$ , effectively increasing the navigable space in the XZ plane by a factor of  $k \cdot k$  (see Figure 5.2(A-B)). Finally, this scaled navigable volume is mapped to a specific part of the virtual world using a constant transformation  $\mathbf{M}_W^T$  (i.e., teleporting can be implemented by dynamically modifying  $\mathbf{M}_W^T$ ).

$$\mathbf{M}_W^U(t) = \mathbf{M}_W^T \cdot \mathbf{S}(\mathbf{k}, \mathbf{1}, \mathbf{k}) \cdot \mathbf{M}_T^U(t) \quad (5.1)$$

This mapping is invertible, showing a bijective mapping between spaces **W** and **T**:

$$\mathbf{M}_T^U(t) = \mathbf{S}(\mathbf{k}, \mathbf{1}, \mathbf{k})^{-1} \cdot (\mathbf{M}_W^T)^{-1} \cdot \mathbf{M}_W^U(t) \quad (5.2)$$

In contrast, our approach relies on the previous position of the user's head and is not directly invertible. At each point in time, the current position is computed from the previous virtual position  $\mathbf{M}_W^U(t)$  and the current real displacement of the user's head (Eq. (5.3)). This displacement is scaled by a variable factor  $\mathbf{k}(\mathbf{M}_W^U(t))$ , which depends on the location of the user in the virtual world (Eq. (5.4)). Orientation (direction) of motion is not affected. This function relating the virtual location of the user to displacement represents our *navigation field* and is explained in the following section, being a key element for the adaptive nature of our technique.

$$\mathbf{M}_W^U(t + dt) = \mathbf{M}_W^U(t) + \mathbf{D} \left( \mathbf{k}(\mathbf{M}_W^U(t)) \right) \cdot \frac{d\mathbf{M}_T^U(t)}{dt} \quad (5.3)$$

$$\mathbf{D} \left( \mathbf{k}(\mathbf{M}_W^U(t)) \right) = \mathbf{S}(\mathbf{k}(\mathbf{M}_W^U(t)), \mathbf{1}, \mathbf{k}(\mathbf{M}_W^U(t))) \quad (5.4)$$

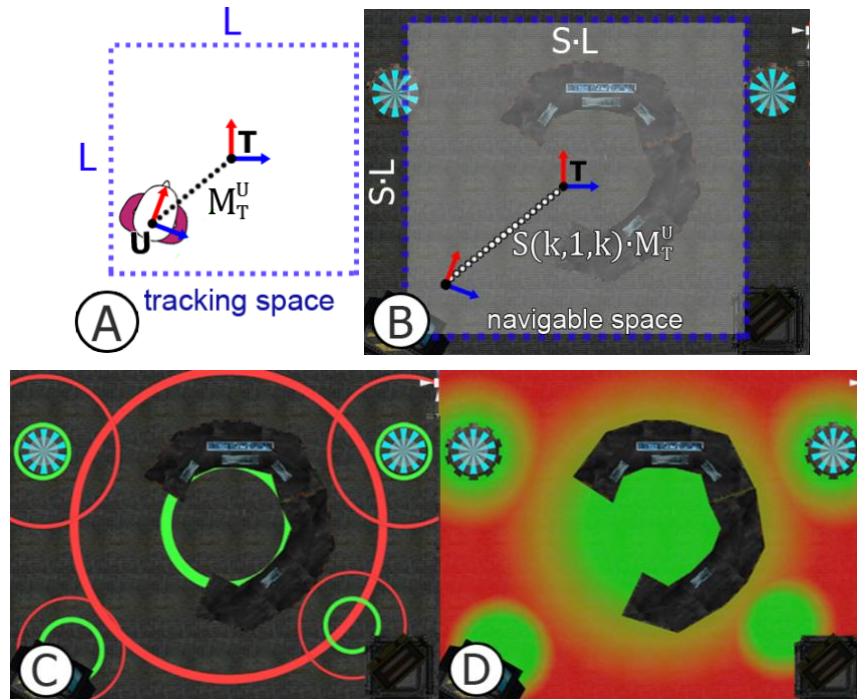
The initial location of the viewpoint is defined as in Eq(5.5), mapping the navigable volume to a specific part of the VE:

$$\mathbf{M}_W^U(0) = \mathbf{M}_W^T \cdot \mathbf{M}_T^U(0) \quad (5.5)$$

### 5.3.2 Generating the Navigation Field $k(\mathbf{M}_W^U(t))$ :

The previous section described the adaptive technique enabling variable displacement according to the user position in the VE. However, it is still necessary to define the *navigation field*, that is the scalar field  $k(\mathbf{M}_W^U(t))$  describing the scaling factor to apply at each point of the VE.

Being a scalar field, navigation fields can be represented as textures, with the value of each pixel describing the scaling factor to apply (and we will represent them as such throughout the chapter). However, here we describe a general approach to automatically compute this *navigation field* based on a set of interactive areas. To do so, we first model the contribution of each interactive area and then compute the final field from these individual contributions.



**Figure 5.2 NaviFields' principle:** (A) Tracking spaces are usually reduced. (B) Physical displacement with homogeneous scaling can increase the navigable volume by a constant factor  $S$ . (C) In NaviFields, each interactive area provides a different scaling function, based on how relevant each point is to interact within that area. (D) We compute the final Navigation Field, by combining these individual contributions.

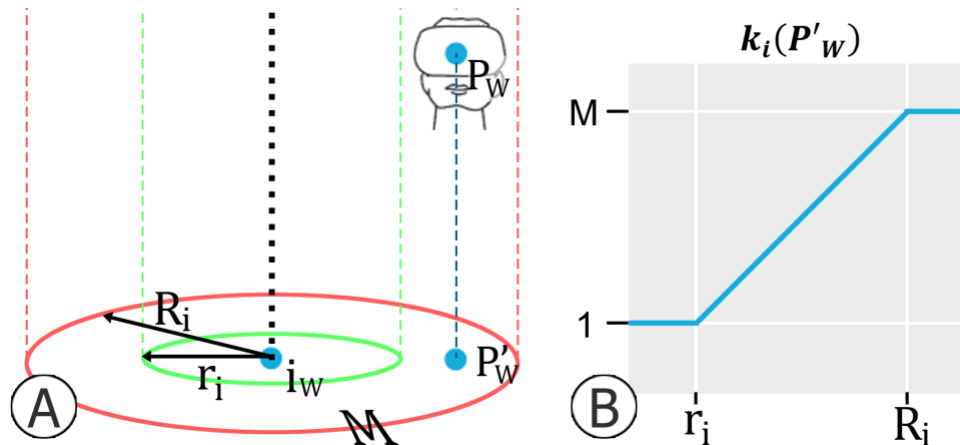
### 5.3.2.1 Per Interactive Area Contribution:

For each interactive area, we compute a simple function that describes the scaling to be applied to a user position, based on how relevant that position is to interact in that area. Let  $\mathbf{I}$  be our set of interactive areas in the world ( $\mathbf{W}$ ). We model each area  $\mathbf{i} \in \mathbf{I}$  as a tuple  $\mathbf{i} = \{i_w, r_i, R_i, M\}$ , representing two concentric cylinders of radii  $r_i$  and  $R_i$  centered around  $i_w \in \mathbb{R}^4$ , and with maximum scaling factor  $M$  (see **Figure 5.3(A)**).

Let  $\mathbf{P}_w = \mathbf{M}_w^U(t) \cdot (0,0,0,1)^T$ , be the current position of the virtual head and  $\mathbf{P}'_w$  its projection on the horizontal plane ( $Y=0$ ). We define contribution of area  $\mathbf{i}$  to the field as in Eq. (5.6), where  $d = \|\mathbf{P}'_w - \mathbf{i}_w\|$  represents the distance between the user and the centre of the interactive area:

$$\mathbf{k}_i(\mathbf{P}'_w) = \begin{cases} 1 & , d \leq r_i \\ 1 + (M - 1) \cdot \frac{d - r_i}{R_i - r_i} & , r_i < d \leq R_i \\ M & , R_i < d \end{cases} \quad (5.6)$$

This function provides no scaling ( $\mathbf{k}_i(\mathbf{P}_w) = 1$ ) inside the inner cylinder, to facilitate maneuver and precise interaction. The scaling factor increases linearly between the inner and the outer cylinders, to ease navigation to distant points. It must be noted that, although this function shows a linear behaviour, it operates on the user's velocity ( $d\mathbf{M}_T^U(t)/dt$ , in Eq. (5.3)). Thus, user moving away from  $\mathbf{i}$  at constant speed will actually experience a parabolic motion. This is inspired from related approaches of viewport control (Mackinlay et al., 1990), to reduce simulation sickness (López, 2013) and maintain spatial awareness (Bowman et al., 1997).



**Figure 5.3 Modelling navigation fields: (A) Interactive areas are defined by their inner and outer cylinders (radius  $r_i$  and  $R_i$ ) and their maximum scaling factor  $M$ . (B) Function described by each interactive area.**



### 5.3.2.2 Global Navigation Field:

Each area  $i \in I$  provides a different scaling function, based on how relevant each point is to interact with that area. Thus, for given a point  $P_W$ , each area will provide a different scaling factor. We resolve conflicts by describing the *navigation field* as the minimum scaling factor across all interactive areas (see Eq. (5.7)). This ensures natural navigation inside all interactive areas ( $k(P_W) = 1$ ). User motion will also be speed up when leaving an interactive area and slowed down again when arriving to a new area.

$$k(P_W) = \min\{k_i(P_W)\}, i \in I_W \quad (5.7)$$

It must be noted that all the definitions provided in this section only scaled displacements along the horizontal plane (assumed XZ). This is convenient for most indoor VEs and avatars resembling humans. In other application contexts (e.g., a spaceship game, where head physically controls motion of the ship) scaling factors  $k$  and  $k(M_W^U(t))$  should also affect the Y coordinate. Similarly, volumetric textures should be used to represent these such *navigation fields*.

## 5.4 User Study

The previous sections motivate the need for *NaviFields* and provide a formal definition for the technique. In this section we assess the usability of the navigation technique for search travel and manoeuvring (low-level navigation tasks (Bowman et al., 2001)).

We compare our technique to the use of homogeneous scaling (Williams et al., 2006), rather than other scaled physical displacement approaches (LaViola Jr et al., 2001; Freitag et al., 2016), as these later ones impose an egocentric approach that does not match the inherent *allocentric* nature of *NaviFields* (i.e., scaling controlled by the environment, not the user). Rather than focusing the study on egocentric vs *allocentric* navigation, we include an additional comparison to natural navigation. This baseline comparison, and the extensive analysis of the factors influencing search and manoeuvring (scaling factors, path lengths, manoeuvring complexity and user poses), allow us to present a full testbed evaluation on the particularities of the technique proposed.

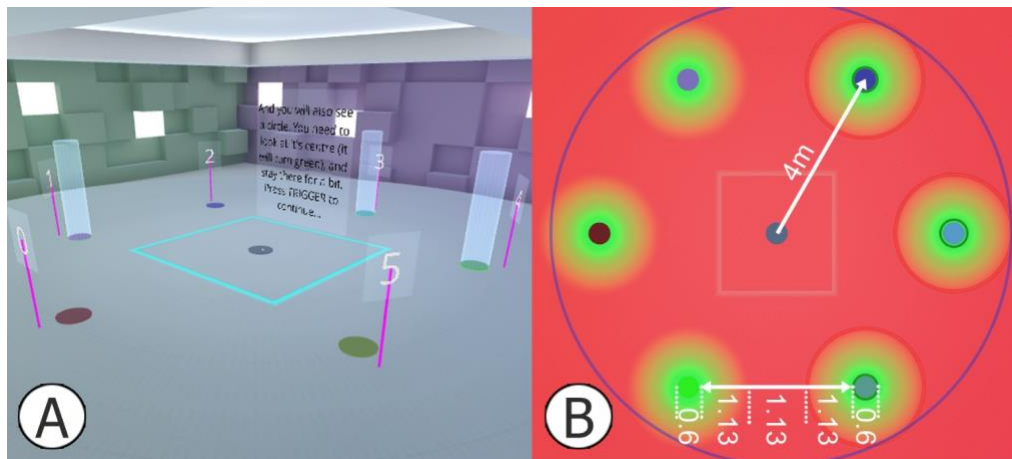
### 5.4.1 Participants

We performed our study across two different European countries (Spain and UK). Both locations used equipment with similar performance (90fps) and an empty experimental space of 3x3m for navigation. We used written-in-game instructions in both languages to guide participants' training, to minimize differences across countries (i.e., different

experimenters providing the instructions) and to reduce bias due to such oral instructions. Following this process, we recruited 24 participants (12 in each country), collecting written consent from them before the start of the experiment.

#### 5.4.2 Testing Environment and Navigation Tasks

We implemented our technique<sup>1</sup> using Unity, and an HTC Vive headset, and created a testbed environment to evaluate it (**Figure 5.4(A)**). The environment contained six target areas of 60cm Ø, identified with numbered flags and evenly distributed around the user's initial position. Participants were invited to stand in the centre of the experimental space, wearing the headset and holding one of the HTC controllers. At the beginning of the experiment, users went through a short in-game walkthrough, to familiarize then with the virtual environment and the tasks. They also performed one travel task (using natural navigation) and one manoeuvring task (see the next section) for training.



**Figure 5.4 Virtual environment: (A) Screenshot of the test environment implemented, with six target flags around initial location. (B) The navigation field was computed for each condition, and used to compute the equivalent homogeneous scaling factor.**

For the following trials, participants performed a two-part task, the first one to test the usability of the techniques for travelling, and the second one focused on manoeuvring.

##### 5.4.2.1 Travelling Task:

At the beginning of each trial, users were positioned in the centre of the tracking space (and aligned to the centre of the VE). A text box was then displayed in front of them, describing a sequence of flags they had to travel to (target flags). Target flags appeared highlighted (see **Figure 5.4(A)**). Participants were allowed to look around to identify the flags (for planning and wayfinding). When ready (and only if still standing inside the central area) users pressed the trigger on the controller, to start the task and travel towards the flags. The task finished when the user reached the final flag.

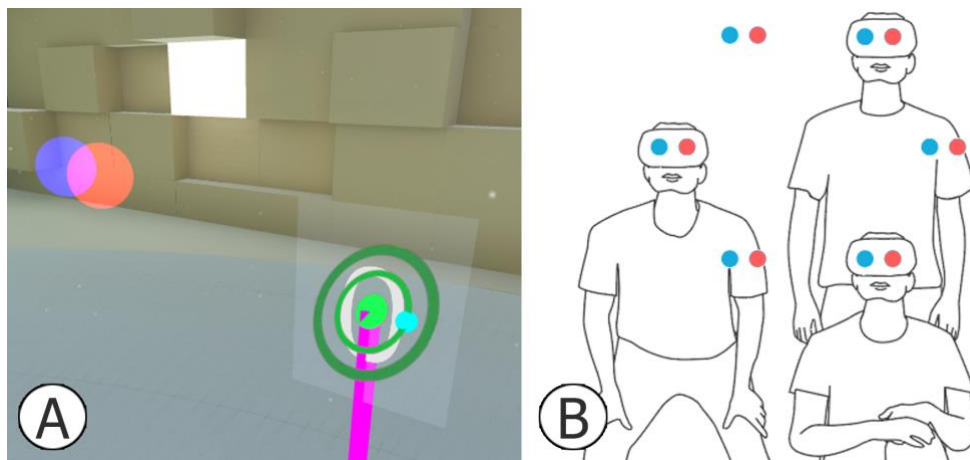


#### 5.4.2.2 Manoeuvring Task:

When participants reached the last flag in the travelling task, an audio signal notified them of the start of the manoeuvring task. We then used an adapted version of in-world ParaFrustum (Sukan et al., 2014) (see **Figure 5.5(A)**) to describe the manoeuvring task. Participants had to attain and keep a correct head position and orientation for one second, to complete the trial.

As defined in (Sukan et al., 2014), ParaFrustum is “a geometric construct that represents this set of strategic viewpoints and viewing directions and establishes constraints on a range of acceptable locations for the user’s eyes and a range of acceptable angles in which the user’s head can be oriented”.

In our adaptation, two spheres (red for the left eye and blue for the right eye) showed to the users where they had to position their eyes (size of the spheres reflected the positioning tolerance allowed). After positioning, a green ring (tail/target, in the ParaFrustum notation) identified where users had to look at. A small cursor helped users to align their view to the target. The size of the ring was computed based on the maximum orientation error allowed. Thus, if the cursor was inside the ring, the orientation error would be small enough. Please note that this differs from the original ParaFrustum proposed in (Sukan et al., 2014), where the ring is shown at the periphery of the vision field. While this might be appropriate for the field of view (FoV) of the device they used (~60 deg), the wider FoV in the Vive pushed us to use this alternative implementation (i.e., to keep attention focused on the target to look at, instead of on the periphery of vision).



**Figure 5.5 ParaFrustums: (A) Screenshot of the in-world ParaFrustum implemented. (B) ParaFrustums were randomly located among 6 potential locations, each enforcing a different pose of the user**

In each manoeuvring trial, six ParaFrustum's were displayed in six potential positions around the flag, arranged in a hexagonal pattern (see **Figure 5.5(B)**). These forced three different poses in the users: kneeling (maximum stability, but reduced mobility), medium (low stability, higher mobility) and standing (good stability and mobility).

#### 5.4.2.3 *Self-report:*

At the end of each trial, participants were asked to answer 4 questions. Two inquiring how easy/comfortable it was to walk to the flags (i.e., travel task) and two inquiring how easy/comfortable it was to look at the targets (i.e., manoeuvring task). Questions were displayed either in English or in Spanish on floating textboxes. Each question used a Likert scale from -3 to 3, which participants selected by moving the controller on their choice and pressing trigger.

### 5.4.3 Experimental Design

#### 5.4.3.1 *Travelling Task:*

In the experiment, we compared the proposed *NaviFields* technique (NF) to Physical Displacement with a homogeneous scaling factor (PH). We adopted a 2x3x2 full factorial design, with these factors being the condition tested ( $T=\{NF, PH\}$ ), the scaling factor used ( $S=\{2, 4, 8\}$ ), and the lengths of the travelling path (between  $L=3$  or  $L=5$  flags, see travelling task above). Scaling factors were selected based on a pilot study with 8 users, where  $S=16$  was found too high.

Experimental trials were pseudo-randomized using Latin Squares, counterbalancing order among participants according to technique, scaling factor and path length. That is, trials were presented within 2 blocks of six trials each (one block per NF or PH condition). These six trials then counterbalanced scaling factors ( $S=\{2, 4, 8\}$ ,  $S=\{4, 8, 2\}$  or  $S=\{8, 2, 4\}$ ) and path lengths ( $L=\{3, 5\}$  or  $L=\{5, 3\}$ ).

After the two condition blocks, participants completed a third block with natural navigation ( $S=1$ ), which was used as a control condition (most natural and with most experience from the user). This will allow us to express our results as a deviation from a baseline, comparing performances in PH and NF as deviations from the optimal/baseline condition.

Using this approach, we maintained our experimental design balanced and fully factorial. An alternative would have been to add a fourth scaling factor ( $S=1$ ) to the design. However, when  $S=1$ , NF and PH behave in the same way. Thus, trials with  $S=1$  would

be performed twice more than any other trial. Also, removing  $S=1$  in either NF or PH would have resulted in an unbalanced experimental design.

Finally, we made sure that, while the starting flag for each travelling task was randomly selected, the total distance to travel between flags was always equal for all trials under the same  $L$  condition. In other words, all paths with  $L=3$  had the same length, as well as all paths with  $L=5$ . Up to 24 paths of equal distance were identified for  $L=3$ , and 108 paths for  $L=5$ , with paths being randomly selected for each trial.

#### 5.4.3.2 *Manoeuvring Task*

The manoeuvring task was performed at the end of each travel task. Given the experimental design explained above, each manoeuvring task was repeated twice for each scaling factor and condition (i.e., for any given  $S$  and  $T$ , there is one manoeuvring trial with  $L=3$ , and second one with  $L=5$ ). This allowed us to test twelve different ParaFrustums per scaling factor and condition. The twelve ParaFrustums were a combination of three different factors, namely the ParaFrustum's position ( $P$ ), head size ( $HS$ ) and tail size ( $TS$ ).

Position ( $P=\{KP, MP, SP\}$ ) was connected to the location of the ParaFrustum (see **Figure 5.5(B)**) and represented the user pose that allowed him/her to reach the eye spheres, either kneeling ( $KP$ ), medium ( $MP$ ) or standing ( $SP$ ).

Head size ( $H_S=\{TH, LH\}$ ) related to the position error tolerance of the ParaFrustum. That is the size of the red and blue spheres indicating to the users where to position their eyes. Two sizes were compared: Tight head ( $TH$ ; +1.5 cm max eye position error) and Loose head ( $LH$ , + 3cm).

Tail size ( $T_S = \{TT, LT\}$ ) related to the orientation error allowed by the ParaFrustum and was visually connected to the size of the target ring. Two sizes were tested: Tight tail ( $TT$ ; max orientation error +5 degree) and Loose tail ( $LT$ , max orientation error +10 degrees).

#### 5.4.4 *Scaling Factors, Environment Size and Navigation Fields*

To test navigation across bigger virtual environments, the distance of the flags to the centre in the virtual environment increased proportionally with the scaling factor ( $S$ ). More specifically, flags were located at  $D_s=\{2m, 4m, 8m\}$  from the centre respectively for each  $S=\{2,4,8\}$ . For the baseline condition (natural navigation) flags were located at  $D_1=1m$  from the centre of the VE.

The configuration of the interactive areas (the flags) also changed according to the scaling condition  $S$ . All areas  $i \in I$  maintained an inner radius  $r_i = 30$  cm (60cm Ø), but their outer radius was set to  $R_i = r_i + (D_s - 2r_i)/3$ . For two adjacent flags, this outer radius covered one third of the distance between the edges of their inner 1:1 areas (**Figure 5.4(B)**) shows an example for  $S=4$ , with specific measurements in meters). Thus, independently of the scaling applied, a user travelling between adjacent flags would go through the transition zone of the first flag for one third of the trip, through the area of maximum speed during the second third, and into the second flag's transition zone in the last third.

Finally, the scaling factors ( $S$ ) needed to be compensated across techniques. For example, a scaling factor  $S=2$  in PH would apply in all the navigable space. However, if we used  $M=2$  in the interactive areas (see Eq(5.6), for the meaning of  $M$ ), this scaling would only apply when users are outside of all interactive areas (i.e., red areas in **Figure 5.4(B)**).

To balance these conditions for each  $S$  factor, the interactive areas will use an  $M$  value that provides an average scaling across the *navigation field* equal to our target value  $S$ . To compute these equivalent  $M$  values, we simply configured the areas as described above (distance to centre  $D_s$ , inner and outer radius values). We then performed a linear search, testing increasing  $M$  values until the average scaling factor (i.e., integral of the *navigation field* divided by area) matched our target value  $S$ . Using this approach the equivalent  $M$  values to use for each  $S \{2,4,8\}$  were  $M=\{2.19, 4.60, 9.17\}$ . Note that these values need to be slightly higher than PH, to compensate for the areas where no scaling is applied.

#### 5.4.5 Usability Evaluation Criteria

The experimental software automatically recorded several dependent variables. For the travelling trials these were: task completion time ( $T\_TCT$ ), real distance travelled ( $T\_RD$ ) and deviation ( $T\_D$ ).  $T\_TCT$  measured the time since the participants arrived to the first flag until they reached the final flag.  $T\_RD$  measured the distance users moved their head (in reality). Finally,  $T\_D$  measured the ratio between length of the virtual trajectory followed (linear integral along the path) divided by the optimum/minimum path length.

For manoeuvring, the variables recorded were: task completion time ( $M\_TCT$ ), number of fixation attempts ( $M\_FA$ ), average position error ( $M\_PE$ ) and average orientation error

(M\_OE). M\_TCT measured the time required to complete the task. The task required users to stay within the constraints of the ParaFrustum continuously during one second. Leaving it, even for one instant would reset our one second timer. The number of fixation attempts (M\_FA) counted the number of times this happened. Average position error (M\_PE) and average orientation error (M\_OE) measured error only while the user was correctly located within the ParaFrustum constraints.

Finally, we collected the responses to the four questions as Q\_CT, Q\_ET, Q\_CM and Q\_EM, to refer to the ease (E) and comfort (C) for the travel (T) or manoeuvring task (M)

## 5.5 Results and Partial Discussion

In this section, we report the results of our user study. Independent sample t-tests across countries showed no significant differences. During our joint analysis of travel, manoeuvre and self-reports, we used within-subjects analysis of variance (ANOVA) to analyse the impact of each factor on the dependent variables (explained above). We also measured interactions between technique (T) and the other independent variables in the task (e.g., interactions between condition T and variables L and S, for the travelling task).

Please note that the reported values (average and standard deviation) are reported as **deviations from baseline (natural navigation)**. Given the high number of features examined, and because of the nature of the results (many significant interactions between variables), we will only report the absolute results in [Appendix 2](#). Where needed, post hoc analysis with Bonferroni corrections were performed and most of them can be found in **Figure 5.6** and **Figure 5.7** (i.e., significance between pairs is indicated with an asterisk; and the difference between the pairs can be assessed from the graphs itself). The average and standard deviation of the baseline is also included in a box under the horizontal axis of each graph (i.e., to help assess the relevance of the effects observed between conditions). Numerical reports, absolute averages and standard deviations (instead of deviation from baseline) and average and standard deviation indicating the exact values of factors in the significant interactions can be found in [Appendix 2](#).

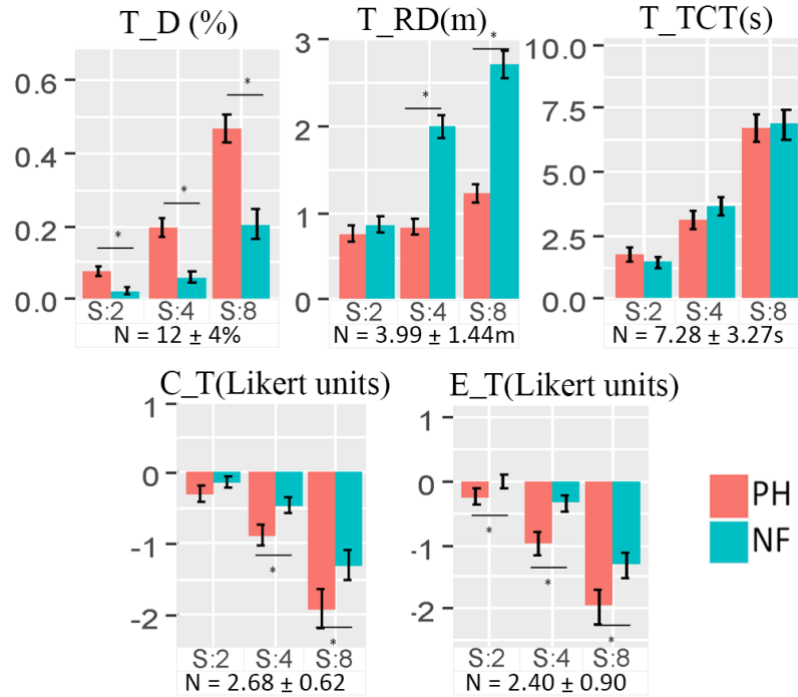
### 5.5.1 Traveling task: NaviFields vs. Homogeneous Scaling

ANOVA results for travelling are shown in **Table 5.1**, both for the main effects and their interactions with T. Both techniques behaved on average worse than the baseline (see **Figure 5.6**) in terms of deviation from the optimal path (T\_D). Lateral head movements

while walking could justify this, as these were scaled by both techniques. Even if the torso was moving linearly (following the optimum path), the sinusoidal side movement of the head would be scaled, resulting in higher distance travelled and worse T\_D.

	T	S	L	T*S	T*L
<b>T_D</b>	F=29.53 p<0.001	F=43.45 p<0.001	F=3.04 p=.093	F=8.13 p<0.001	F=0.45 p=0.50
<b>T_RD</b>	F=111 p<0.001	F=98.62 p<0.001	F=78.74 p<0.001	F=78.68 p<0.001	F=23.44 p<0.001
<b>T_TCT</b>	F=0.229 p=0.63	F=76.96 p<0.001	F=73.9 p<0.001	F=2.10 p=0.13	F=0.22 p=0.63
<b>CT</b>	F=6.646 p<0.05	F=32.03 p<0.001	F=0.404 p=0.531	F=1.182 p=0.315	F=0.026 p=0.87
<b>ET</b>	F=15.8 p<0.001	F=31.75 p<0.001	F=2.229 p=0.14	F=0.904 p=0.412	F=0.036 p=0.851

**Table 5.1 Results from repeated measures ANOVA on travel-related features and questionnaire ratings (CT and ET rows). Effects of technique (T), scaling (S) and path (L), as well as interactions among them (T\*S, T\*L)**



**Figure 5.6 Results of traveling task: Bar plots representing mean and standard error of the variables collected in the travelling task, for each condition and scale factor. Significant post-hoc tests (p<0.05) between condition at each level of scaling are marked with \*.**

However, our technique (NF) showed much lower deviation than homogeneous scaling (PH), with this difference becoming more relevant for higher scaling factors (significant interaction T\*S). Paired analysis (indicated with asterisks in **Figure 5.6**) reveals differences between NF and PH are significant under all scaling conditions. This could be the result of head side displacements being scaled less while the user travelled through

low-scaling parts of the virtual environment (inner areas and transition zones). It is also worth noting that NF was only significantly worse than the baseline for  $S=8$  (paired t-test, Bonferroni corrected,  $p<0.05$ ), indicating that for lower factors, users could still follow their paths effectively.

On average, users also moved more in the real environment (T\_RD) with NF and PH than in the baseline condition (see **Figure 5.6**). The difference between PH and baseline can be explained by looking at our users' behaviour. Paths were equivalent for PH and baseline if users passed through the centre of the interactive areas. However, as soon as users reached the inner radius of the area, a sound was triggered, and most users directly proceeded to the next flag (travelling between the *edges* of the areas). Thus, using PH with scaling  $S=4$ , the 60cm interactive area would be reduced to 15cm in the real world (with users traveling 22.5 cm more to reach *edges*). Considering the distances between flags from **Figure 5.4(B)**, this led to a final travel distance of  $339\text{cm}/4 = 85\text{cm}$ .

This situation was even worse for NF. Using the same example (and measurements from **Figure 5.4(B)**), users would require 40cm to travel through both transition areas, and 25 cm to go through the area of maximum scaling, resulting in a required total distance of 105cm. These differences in the distances required for each technique actually increased with higher scaling factors, further penalizing the NF condition.

The fact that participants moved more in the real environment in both NF and PH can also help explain why users took more time to travel (T\_TCT) in the two conditions (see **Figure 5.6**), and the influence of  $S$  and  $L$ .

However, we found the lack of significant differences between NF and PH in terms of T\_TCT interesting. We actually expected *NaviFields* to behave worse than PH, given that: **a)** the user was not in control of the velocity (depends on his position in the VE); and **b)** bigger  $S$  and  $L$  values should have penalized the NF technique even more than PH in terms of T\_RD. The most likely way NF users could cover more real distance in (not significantly) higher time, would be if they actually moved faster in reality, which would indicate a higher level of confidence during locomotion.

Questionnaire results also aligned in this direction, showing a general preference for NF (significance of T, for both C\_T and E\_T), which was further reinforced at higher scaling factors and distances (interaction  $T*S$ , and  $T*L$  for both C\_T and E\_T). Particularly, for all factors higher than  $S=2$ , travel was considered easier in NF and also more comfortable

(see significant post-hoc tests in **Figure 5.6**, for C\_E and E\_T). The fact that human motor control is planned in advance, based on the information collected from the environment (Kawato, 1999; Hayhoe et al., 2009), could also influence the better performance and preference for NF. Despite both NF and PH scaling users' movements, we observed participants using NF tended to look around to plan the travel trajectory before starting locomotion (e.g., while inside a 1:1 area). The progressive increase in scaling could also have helped them to tune and adapt motion, once travelling started. These factors could help participants in the NF condition perceive travel as more comfortable. Finally, the progressive slow down when reaching the flags could serve as a feed-forward, reassuring them on the successful completion of the task, before actually finishing it.

	T	S	H_S	T_S	P
M_TCT	F= 185.7 p < .001	F=103.7 p< .001	F= 09.7 p< 0.001	F=5.332 p < 0.01	F = 10.9 p< .001
M_OE	F=41.27 p<.001	F=19.77 p<.001	F = 1.35 p = 0.25	F =9.282 p < 0.001	F=0.295 p = 0.59
M_PE	F= 12.99 p < 0.01	F= 7.39 p< 0.01	F= 5.866 p< 0.05	F= 0.115 p = 0.07	F= 0.017 P = 0.89
M_FA	F= 267 p < 0.001	F=134.8 p< 0.001	F=97.81 p< 0.001	F = 0.629 p =0.43	F = 9.586 p < 0.05

**Table 5.2 ANOVA effects for each independent variable.**

	T*S	T*H_S	T*T_S	T*P
M_TCT	F= 140.2 p< 0.001	F=97.25 p< 0 .001	F= 2.54 P = 0.124	F=13.59 p< 0 .01
M_OE	F=17.5 p <0.001	F=5.64 p <0.05	F=0.64 p =0.4	F=0.002 p = 0.96
M_PE	F= 5.08 p< 0 .001	F= 2.20 P=0.1	F= 6.459 p< 0.05	F= 7 p< 0.05
M_FA	F = 174.3 p< 0.001	F = 125.3 p< 0.001	F = 2.02 p = 0.16	F = 8.5 p < 0.05

**Table 5.3 ANOVA effects for the interactions of the factor T with dependent variables for manoeuvring**

	T	S	L	T*S	T*L
CM	F= 82.84 p< 0.001	F= 47.18 p< 0.001	F= 3.194 P= 0.08	F= 77.33 p< 0.001	F= 3.135 P=0.08
EM	F= 60.15 p< 0.001	F= 39.73 p< 0.001	F= 0.59 p= 0.44	F= 42.97 p< 0.001	F= 4.17 P = 0.052

**Table 5.4 ANOVA effects for questionnaire questions and interactions of factor T with each of the remaining variables.**

### 5.5.2 Manoeuvring: NaviFields vs. Homogeneous Scaling

**Table 5.2** reports the main effects from the ANOVA model for the independent variables, while **Table 5.3** reports the main effects for all the interactions. **Table 5.4** reports ANOVA results for the manoeuvring questionnaires.



Our results clearly show *NaviFields* (NF) provided better results than PH for manoeuvring. Significant effects were found for all variables (with NF always performing better). Also, higher scaling factors increased these differences even more (note the significant interactions T\*S according to all variables; diverging trends clearly observable in **Figure 5.7**).

These results were expected. For instance, in the case of a small ParaFrustum head (3cm) and S=4; participants using PH had to place their head within a sphere of 0.75cm. When compared to baseline, PH was significantly worse for S factors bigger than 2 ( $p < 0.05$  for all post-hoc tests between scaling and baseline; see [Appendix 2](#)).

In contrast, NF enables similar amount of control than natural navigation. Indeed, when comparing NF to the baseline, no significant effect of scaling or condition was found for any of the variables, including questionnaires (CM and EM). Although expected (NF allowed close to 1:1 navigation during manoeuvring), these results clearly illustrate one of the main strengths in NF (allow larger navigable spaces, but still allow precise manoeuvring), and also show the impact scaling can have on manoeuvring tasks.

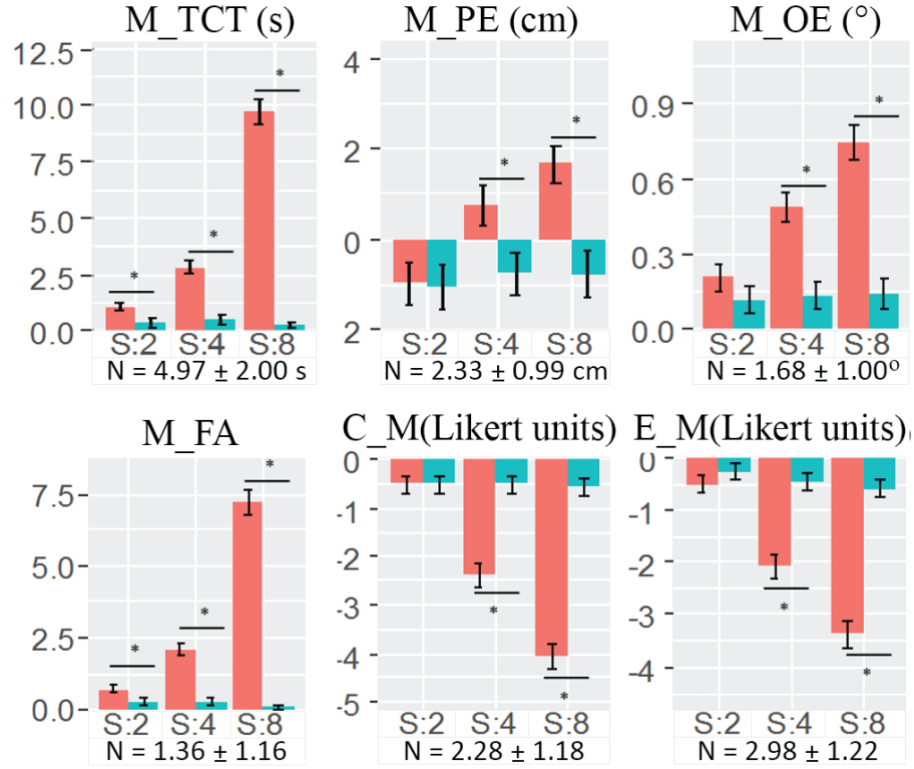
Besides, our study also revealed some other relevant aspects related to manoeuvring tasks, which can be applied to *NaviFields*, or any other physical displacement navigation techniques scaling user's motion ([Song et al., 1993](#); [Williams et al., 2006](#); [Interrante et al., 2007](#); [Steinicke et al., 2010](#)).

First, these results seem to challenge related work, where a homogeneous scaling factor of S=10 was regarded as still comfortable for users ([Williams et al., 2006](#)). Results from both travel and manoeuvring show a clear decrease in performance as scaling factors increase. The precision demands of the task also have a great influence on this factor (i.e., more significant effects in our manoeuvring results).

However, it was interesting to observe that users seemed to have a relatively good manoeuvrability with a scaling factor of 2, both for NF and PH. In this condition (S=2), no significant differences could be observed between PH and NF (or baseline) in terms of M\_OE and M\_PE (observe lack of differences between pairs in **Figure 5.7**, for S=2). Questionnaires further reinforced this observation: for S=2 no significant difference were found between NF and PH ( $t < 1$  and  $p > 0.05$ ), for both CM and EM.

This seems to indicate that, even in precise tasks, users can adapt their movement to finely control their gaze and posture, even in conditions where their visual feedback is

dissociated from their proprioceptive and vestibular feedback by a factor of 2. This could encourage the use of scaling factors bigger than one even inside relevant areas. This could further increase the additional navigable space while not affecting interaction significantly.



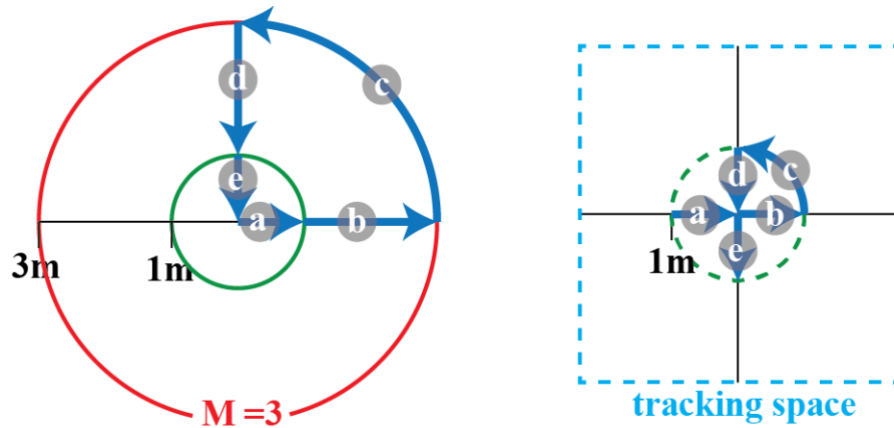
**Figure 5.7 Results of manoeuvring task:** Bar plots representing mean and standard error of the variables collected, for each condition and scaling. Significant post-hoc tests ( $p < 0.05$ ) between condition at each level of scaling are marked with \*. Since our plots depict the deviation from the baseline condition (natural navigation), the mean and standard deviation are presented under each graph as "N = Mean,  $\pm$ SD".

Our results also seem to indicate that, when scaling was applied, manoeuvring complexity was mostly driven by the position error allowed by the ParaFrustum (rather than the orientation error). It is worth noting that ParaFrustum head sizes chosen ( $H_S = \{3\text{cm}, 6\text{cm}\}$ ) had a significant effect on M\_PE (less error for loose head sizes). The same applied to tail size ( $T_S = \{\pm 5^\circ, \pm 10^\circ\}$ ), with main effect on M\_OE (again, less error for loose tail sizes). These results indicate that the sizes and angular ranges chosen actually represent two positioning and two orientation tasks with different levels of complexity. However, the influence of  $H_S$  (Effect size on M\_OE, Choen's  $d=0.23$ ) was much bigger than  $T_S$  (Effect size on M\_OE, Choen's  $d=0.5$ ). This relevance of positioning vs orientation should be specially considered when designing tasks for points of the VE with higher scaling factors (for  $S > 2$ , allow more positioning error).

Finally, the user's pose had significant effects on the time ( $M\_TCT$ ) and number of attempts required ( $M\_FA$ ). Trials completed standing had on average the best performance and, surprisingly, kneeling led to the worst performance (higher  $M\_PE$ ,  $M\_TCT$ ,  $M\_FA$ ,  $M\_OE$ , when compared to other poses). This seems to indicate mobility range can become a much more relevant factor than stability, for manoeuvring in environments using displacement scaling. We observed that, the small movement of the participants' head while kneeling (down, but also forward) was scaled up, and users would tend to move past their target location. Users had to learn and anticipate this, either avoiding forward motion while kneeling, or by kneeling at a further distance to the target. This made it more difficult to reach the desired position, and the more limited range of motion of the pose, also offered less chances to correct it. Thus, users needed several attempts before "landing" in the correct spot.

### 5.6 Drifting Effects in NaviFields

Another observed effect from our studies was the presence of *Drift* in our technique (this effect is more deeply studied and corrected in the next chapter). Briefly, the effect occurs when a user returning to the centre of the VE, does not actually end up in the same position where he started in the real world. Being an unforeseen effect, our software did not collect data as to allow us to provide an empirical assessment of its impact. However, this effect did not result in any major issues during our experiments.



**Figure 5.8 Example of the role of *Drift*: A user walking along a closed path in the VE, will not return to the same real point.**

The effect of *Drift* can be exemplified by **Figure 5.8**, left. This shows a user walking in a closed trajectory near an interactive area with  $r_i = 1\text{m}$ ,  $R_i = 3\text{m}$  and maximum scaling  $M=3$ . In reality (**Figure 5.8**, right), this user would walk 1m across the inner area (arrow *a*) and 1 m across the transition area (arrow *b*; average scale of two). Due to scaling, the

curved trajectory in (c), would require an arch in reality of only  $3/3=1$  m of radius, but in the path back to the centre (d and e, similar to b and a), our example user would end up 1.41 m away from the starting point.

This effect is the result of the different scales used for motion in the virtual and real world. For any given closed path A, the *Drift* vector can be computed as in Eq(5.8) (a full derivation for Eq(5.8) is available in [Appendix 1](#)).

$$Dr(A) = \int_0^A \mathbf{S}(\mathbf{k}(\mathbf{M}_W^U(t) - 1, 0, \mathbf{k}(\mathbf{M}_W^U(t) - 1)) \cdot \frac{d\mathbf{P}_T(t)}{dt} \cdot d\mathbf{t} \quad (5.8)$$

Once modelled, this effect can be addressed by borrowing approaches from redirected walking. For instance, the *Drift* accumulated by the user since the beginning of the session is implicitly represented by the difference between its virtual and real positions ( $\mathbf{M}_W^U(t)$  and  $\mathbf{M}_T^U(t)$ ). However, the effect of the *Drift* cannot be assessed until the user returns to the original position (he/she closes the path). One simple approach is to, at every point in time  $t$ , compute the *Drift* that would be present if the user wished to return to the starting position, (following a linear path L from  $\mathbf{M}_W^U(t)$  to  $\mathbf{M}_W^U(0)$ ):

$$Dr(A') = (\mathbf{M}_W^U(t) - \mathbf{M}_T^U(t)) + Dr(L) \quad (5.9)$$

This estimation can now be used to iteratively reduce *Drift*. If the current displacement (Eq(5.4)) will increase the magnitude of the *Drift* vector, one could positively scale this displacement (i.e., so that the user continues to move in that direction for as short as possible). Alternatively, if the current displacement will reduce *Drift*, user motion should be damped (i.e., to force motion in *Drift*-correcting directions).

## 5.7 Discussion

The results from our experiment have shown very good potential of *NaviFields* as a navigation technique, allowing users to navigate environments up to 8x8 times bigger than natural navigation, with very good potential for both travelling and manoeuvring. However, its effects on higher level aspects on navigation (spatial orientation, way-finding, presence, cyber-sickness) should still be assessed.

Other aspects revealed by our study also deserve further exploration. Reusing models to predict body motion from head motion ([Slater et al., 1995](#); [Cirio et al., 2009](#); [Freitag et al., 2016](#)), could avoid scaling head's lateral motion. *Drift* correction techniques should also be tested.

*NaviFields*' ability to extend the navigable space will also be heavily influenced by the nature of the VE. *NaviFields* will perform well in VEs with a discrete set of relevant areas, spread throughout the space. However, it will degrade to behave like homogeneous scaling if all points of the VE have similar relevance. This can be judged by looking at the gradient of the *navigation field*, as shown in **Figure 5.2(D)** or **Figure 5.4(B)**.

Other alternatives to generate the *navigation field* should also be explored. We described a very simple approach to compute the field, based on cylindrical areas and locations fixed at design time. This allowed us to explore the use of *NaviFields* as a general navigation technique, simplifies understandability and might serve as a general approach, but it is by no means the only way to generate such fields.

As shown at several points throughout the chapter, the field can be described as a 2D map showing the scaling factor applied at each point of the VE. Thus, it can be understood as a continuous entity, where the scaling of each point in space can be tailored individually, to adapt to the specific requirements of the VE. The fields could then be automatically inferred, based on the geometry or architectural cues (e.g., doors, alleys, furniture) of the VE.

Alternatively, an open-ended VE (with no a-priori knowledge of which areas are more relevant) could infer this from the user. The VE could initially use homogeneous scaling (i.e., all points in the *navigation field* sharing the same scaling factor). Clustering techniques could then be used, analysing the points of the VE where the user spends more time, to reduce the scaling factor in those areas (i.e., allow more natural navigation) and increase it in the places where the user spends less time. This could inherently support the learning process in training scenarios, allowing trainees to initially explore the whole environment (e.g., build mental models) and gradually provide adapted support for the areas where they need to spend more time (e.g., workspaces).

The creation/modification of the *navigation field* could also become part of the mechanics of a VR game. In titles such as *Gears of War*, players need to advance among trenches, which become the guiding element for their navigation (i.e., advance to the next trench and then focus on shooting, taking cover or reloading). With *NaviFields*, identifying such areas could add an element of strategy to such games. Users should specifically identify strategic spots (e.g., by a gate, behind a crate) and create relevant areas there. Our technique would allow precise interaction in those locations and fast transitions between them (e.g., to run from one cover point to the next one).

Finally, the technique has always made use of scaling factors bigger than one. Smaller factors would reduce user motion and could be used to avoid penetration into objects (e.g., head getting close to a wall). This could also be used to recreate other effects, such as a user stepping on a muddy patch of the floor (or a slippery patch, using a factor bigger than one).

## 5.8 Conclusions

We presented *NaviFields*, a VR navigation technique that computes the relevance of each point of the VE (*navigation field*) and scales user's motion accordingly. This provides areas of natural navigation (1:1), and faster navigation across non-relevant areas, extending the space users can navigate.

We provided a mathematical characterization of the technique, and implemented it for a testbed environment. We then compared *NaviFields* performance for travelling and manoeuvring, comparing it with homogeneous physical displacement and natural navigation.

Our results show that *NaviFields* is a suitable technique to navigate and interact within the virtual environment. *NaviFields* results are comparable to natural navigation in manoeuvring tasks, and only slightly worse for travelling tasks. Moreover, when compared to homogeneous scaling of the environment, *NaviFields* is judged better in both travel and manoeuvring tasks. Our experimental results also provide insightful information for interaction in VR, highlighting the role of user pose, head position and target size in manoeuvring task, and showing that participants can adapt relatively well for scaling factor up to  $S=2$ . We also analysed the drifting effect observed during the user study, provided a formal model for the effect (reusable for other techniques using differential tracking) and identify strategies to correct it. We finally discussed the scope of application of *NaviFields*, based on its observed properties and the affordances that it enables.

## Chapter 6

# Drift-Correction Techniques for Scale-Adaptive VR Navigation

Based on the *Drift* effect involved in *scale-adaptive* navigation found in the previous chapter, and seeking to better exploit the human capabilities by generating more tailored ITes, in this chapter we mathematically modelled the behaviour of the *Drift* effect across different *scale-adaptive* navigation techniques (i.e., the second stage of interaction) (Molina-Masso et al., 2008), considering also different navigation control types (*ego-centric* and *allocentric*). Then, based on the general formulation of the *Drift* effect, computer simulations and user studies, we propose two modelled correction techniques to address this issue aiming to not only leverage the use of physical locomotion in VR navigation but also address detrimental *Drift* effects that produce turning the navigation techniques unusable.

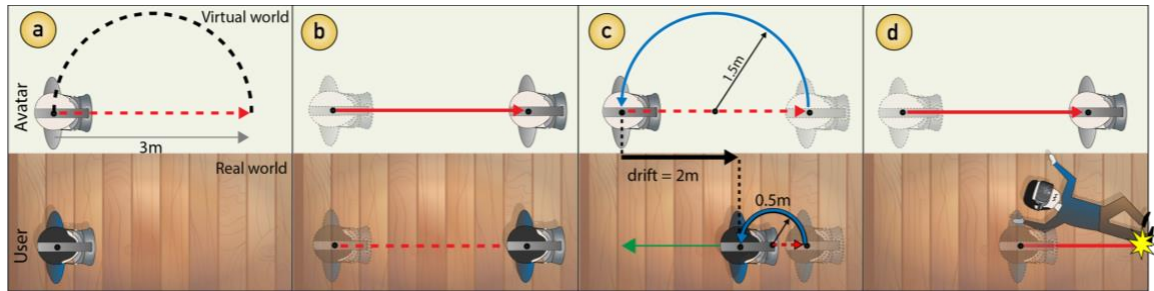
## 6.1 Introduction

Space availability imposes restrictive constraints to the kind of commercial and professional VR experiences created nowadays. Only 9% of VR end users have access to real spaces larger than 3x3m (Community, 2017), which influences the kind of experiences and games created by industry. Natural navigation, arguably the most natural way of locomotion in VR, requires a real space that matches the size of the Virtual Environments (VEs), significantly limiting its current scope of application. Hence, alternative navigation techniques, which do not mimic the way we move and explore in the real world (Anthes et al., 2004), or even might not involve physical displacement at all (Slater et al., 1995) are used extensively.

*Scale-adaptive* navigation techniques could offer a suitable solution to compromise these constraints. Large VEs (e.g., 9x9m) can be accessed even from reduced real spaces (e.g., 2.5x2.5m). Navigation is still controlled by user's physical motion (i.e., closer to natural navigation), although scaled according to various *policies* (e.g., non-linear mapping of speed (Interrante et al., 2007), relevance (as shown in Chapter 5), space boundaries (Song et al., 1993)).

However, this dynamic scaling introduces an issue (*Drift*, exemplified in **Figure 6.1**), that greatly limits their applicability. Due to scaling, the location of the virtual head is not just the result of the current head position in the real world, but the result of its past trajectory of displacements (scaled according to the technique's *policy*). Thus, given an available real space (e.g., 2.5x2.5m), the space that the user will be able to reach in the VE (*navigable space*) will vary according to the past history of user's displacements.

**Figure 6.1** shows an extreme case illustrating the effects of *Drift*. A user follows a closed path (semi-circle) and returns to the origin of the VE (top row, in **Figure 6.1**). The technique (according to an artificial/example *policy*) applies no scaling in the linear path (red, 1:1 ratio) and 3x scaling along the curved part of the path (blue, 1:3 ratio). A user returning to the starting point in the VE would actually end up 2m to the right of the starting position in reality (i.e., see **Figure 6.1(c)**).



**Figure 6.1** The *Drift* effect: (a-c) *Scale-adaptive* techniques apply variable scaling to users' displacements (red=1, blue=3). *Drift* appears as a result, causing that a user following a closed path in VR will not return to the same point in VR (c), which can affect the amount of time that the user can use the technique, before abandoning the limits of the tracking space available (d).

This exemplifies how *Drift* can seriously hinder the practical use of *scale-adaptive* techniques. A VR Designer might have created the VE (i.e., with an intended *navigable space* in mind), and tuned the *policy* of the navigation technique (e.g., max scaling), to initially allow users to reach all relevant parts of the VE. However, *Drift* will accumulate as users navigate, changing the *navigable space* to the point that users might not be able to reach relevant parts of the VE, or even forcing them to stop (i.e., our example user



cannot complete the semi-circular path twice without leaving the real space available). To address this, we propose and test correction approaches that dynamically compute the *Drift* resulting from each user's displacement and adjust the scaling factor to reduce its impact (i.e., minimize displacements when they increase *Drift* and maximize *drift*-correcting displacements).

We first provide a general model to describe *scale-adaptive* techniques and to characterize the *Drift* effect. In a first study, we quantify the effects of *Drift* on several user paths for an example environment and *scale-adaptive* technique. This provides an initial assessment of the ill effects that *Drift* can have on navigation, as well as data to guide the creation and tuning of our correction techniques.

We then present a general model for *Drift* correction techniques and two example implementations: **i)** *Derivative Correction* (based on the proposed model from [Chapter 5](#)), and **ii)** *Angular Correction*, based on our own insight. We then reused the paths recorded from our first study to simulate each technique, systematically exploring varying values of their tuning parameters, observing their effects on *Drift* and identifying an optimum technique and configuration.

We finally evaluate our chosen *Drift*-correction technique with real users, applying it to one *allocentric* (NaviFields: our proposed technique in [Chapter 5](#)) and one *ego-centric* (Seven League Boots: proposed by ([Interrante et al., 2007](#))) adaptive navigation techniques and using a VE different than that in our first study. Our evaluation confirms the effectiveness of our correction for both techniques. *Drift* can be consistently reduced (from values above 0.6m to a steady average value ~20cm). Users can also continue to navigate in the VE for extended distances, before *Drift* pushes them out of the tracking volume (from average 13m without correction to ~69m with our technique), with users not being forced to stop at all in most cases. Finally, our corrections introduce minimum changes to the basic behaviour of the navigation techniques and showed no negative effects in users' performance or their subjective impressions.

## 6.2 Related Work

We reviewed the literature in two interrelated fields: VR navigation techniques and *Drift* effect in VR navigation.

### 6.2.1 VR Navigation Techniques

Physical displacement in immersive Virtual Environments (VE) is commonly constrained by the real available space (or the size of the tracking volume), with natural 1:1 motion requiring a VE equal or smaller than the real space available.

Many metaphors and approaches have been proposed to overcome such limitation (Billinghurst et al., 1997). Some approaches achieve navigation across large VEs at the expense of physical displacements, usually relying on additional controls (e.g., joystick, buttons, wands (Anthes et al., 2004; Williams et al., 2006)). Other techniques rely on users' body motion but with no actual displacement (i.e., walking), such as steering [3, 13, 26] or teleportation techniques (Laurel et al., 1994; Fuhrmann et al., 1998; Bozgeyikli et al., 2016; Liu et al., 2018), but they are usually associated with increased user's spatial disorientation (Bowman et al., 1997; Lathrop et al., 2002). Other methods use walking in place (WIP), where physical displacement is led by the user's simulation of walking (assisted by movement of feet, heels or knees (Slater et al., 1995; Feasel et al., 2008; Wendt et al., 2010)) or treadmills-based systems (Darken et al., 1997; Fernandes et al., 2003; Hale et al., 2014) for more natural locomotion. However, they still lack the same vestibular cues and proprioceptive perception than actual walking (Bowman et al., 2001).

Redirected walking techniques that make use of rotational and translational gains (discussed in Chapter 5, section 5.2.1) provide little benefit for spaces smaller than 6x6m (Azmandian et al., 2015). This is a serious constraint for their practical application, considering that only 9% of non-professional VR users (i.e., end customers) use spaces larger than 3x3m (Community, 2017).

Alternatively, *scale-adaptive techniques* that dynamically adapt user's displacements could be better suited for such tight spatial constraints. They use selection of target destination (Mackinlay et al., 1990), optical flow (Argelaguet, 2014) or head's motion (Interrante et al., 2007) to reduce the physical displacement and induce visual navigation by controlling the viewpoint scale and speed, allowing thus fast displacement along big distances whilst gradually decreasing speed as the user approaches a target. These techniques are usually *ego-centric* (i.e., fully controlled by user motion). Other techniques use an *allocentric* approach where the navigation is based on the VE (Song et al., 1993) i.e., the viewpoint speed is adjusted in specific areas depending of their relevance in the VE (fine or big exploration movements as shown in Chapter 5).

These techniques appear as an interesting alternative, making use of physical displacements (e.g., reduce disorientation, provide proprioceptive perception of walking), allowing navigation of medium-sized VEs and requiring smaller real spaces (e.g., 3x3m as employed in [Chapter 5](#); 3.5x3.5m ([Wilson et al., 2018](#)); 5x5m ([Xie et al., 2010](#))), better suited for the current landscape of VR users, with spaces between 2.5x2.5m and 3x3m being available for ~30% of VR end users ([Community, 2017](#)), and up to ~60% of professional VR spaces ([Community, 2016](#)).

However, *scale-adaptive techniques* also introduce an undesired *Drift* effect (i.e., a difference between the real and the virtual position of the user after traveling along a closed path in the VE), which we describe next.

### 6.2.2 *Drift* Effect in VR Navigation

The *Drift* effect is an inherent problem in VR navigation. We distinguish two main types of *Drift*: hardware-induced and technique-induced *Drift*. The first case (hardware-induced) has been subject to more extensive research ([Siciliano et al., 2016](#)), and is the result from accumulated measuring errors of the device over time (resulting in a progressive mismatch between the position in VR and the real position of users). For instance, inertial sensors have an implicit *Drift* effect due to the nature of the devices used that results in both rotational and translational *Drift* ([Oskiper et al., 2011](#)).

Technique-induced *Drift* is typical from techniques including dynamic modification of user speed (e.g., dynamic scaling factors) ([Interrante et al., 2007](#); [Wilson et al., 2018](#)), as shown in [Chapter 5](#). This results in the same outcome as hardware-induced *Drift* (a mismatch between the “measured” position in VR and the position in reality), but in this case the effect is the result of the accumulated differences between the user’s physical and (scaled) displacements in VR.

Correction techniques have been proposed to deal with hardware-induced *Drift* ([Foxlin, 2005](#); [Oskiper et al., 2011](#)). As per technique-induced *Drift*, studies measure the impact of this effect ([Nilsson et al., 2013](#)), and initial ideas identified in [Chapter 5](#) on how corrections could be implemented. However, to the best of our knowledge, no attempts have been made at correcting technique-induced *Drift*. The correction approach proposed in this chapter uses absolute coordinates (position and orientation of a target with respect to a reference coordinate system) focussing then on technique-induced *Drift* and it is not affected by hardware or sensor deviations (e.g., inertial sensors).

### 6.3 Modelling *Scale-adaptive* Techniques

In this section, we introduce a general model to describe *scale-adaptive* VR navigation techniques, such as (Razzaque et al., 2001; Steinicke et al., 2010; Wendt et al., 2010; Azmandian et al., 2015). This allows us to formalize the *Drift* problem affecting these techniques, which will be instrumental to allow the correction techniques proposed later in the chapter.

For our explanations,  $\mathbf{R}$  and  $\mathbf{V}$  represent the systems of reference of the real and virtual world respectively. Points  $\mathbf{P}_R(t)$  and  $\mathbf{P}_V(t)$  represent the 3D position of the user's head at each point in time (e.g.,  $\mathbf{P}_R(3)$  refers to the real position of the user's head, at time  $t=3s$ ). Capital letters refer to positions in 3D space, while bold lower-case letters refer to vectors (e.g., directions). Operator  $\cdot$  refers to the usual dot product, while  $\odot$  refers to the element-wise (Hadamard) product (Horn et al., 1990).

We also assume spaces  $\mathbf{R}$  and  $\mathbf{V}$  are initially aligned ( $\mathbf{P}_R(0)=\mathbf{P}_V(0)=(0,0,0)$ ). This represents the efforts of the VR designer to identify the *navigable space*, that is, the virtual space that the user should be able to reach considering: **i**) the real space available; and **ii**) the navigation technique chosen. *Drift* effects will affect this initial mapping and can make navigation unfeasible (e.g., areas of the *navigable space* becoming unreachable), as shown later in this chapter.

#### 6.3.1 Uncorrected *Scale-adaptive* Techniques: A General Model

*Scale-adaptive* techniques control the position of the virtual head ( $\mathbf{P}_V(t1)$ ) by applying a variable scaling factor ( $\mathbf{k}(t1)$ ) to the user's real displacements ( $\Delta\mathbf{P}_R(t1)$ ):

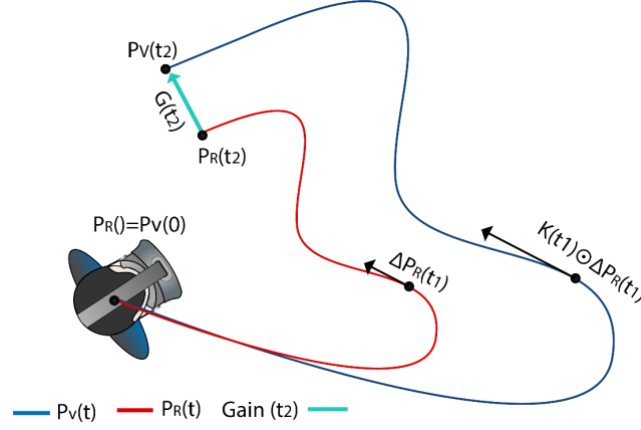
$$\mathbf{P}_V(t1) = \mathbf{P}_V(t1 - \Delta t) + \mathbf{k}(t1) \odot (\Delta\mathbf{P}_R(t1)) \quad (6.1)$$

Function  $\mathbf{k}(t): \mathbb{R} \rightarrow \mathbb{R}_3$ , represents the *scaling policy* of the technique, that is, how the user's movements will be scaled according to the technique's logic. It can be, for instance, the result of a non-linear mapping of the user's speed (Interrante et al., 2007), the distance of the user to the centre of the VR world (Song et al., 1993) or the use of navigation fields explored in Chapter 5. Finally, given our initial assumption ( $\mathbf{P}_R(0)=\mathbf{P}_V(0)=(0,0,0)$ ), real and virtual positions can be defined over time as:

$$\mathbf{P}_R(t1) = \sum_{t=0}^{t1} \Delta\mathbf{P}_R(t); \mathbf{P}_V(t1) = \sum_{t=0}^{t1} \mathbf{k}(t) \odot \Delta\mathbf{P}_R(t) \quad (6.2)$$

### 6.3.2 Translational Gain ( $G(t): \mathbb{R} \rightarrow \mathbb{R}_3$ ):

The *scaling policy* (i.e., modelled by  $\mathbf{k}(t)$  in Eq (6.1)) is key to allow users to navigate spaces different (larger) than the real space. **Figure 6.2** illustrates this behaviour, showing the user's real displacements in  $\mathbf{R}$  and the resulting path in  $\mathbf{V}$ , computed by scaling the user's displacements by  $\mathbf{k}(t)$ .

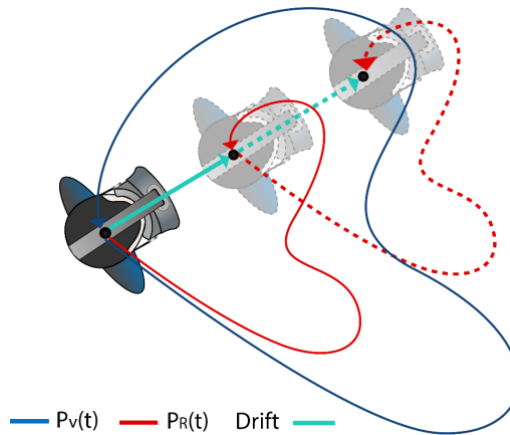


**Figure 6.2** Illustration of user displacement in the real (red line) and virtual spaces (blue line). The green line represents the *Gain* vector (extra displacement enabled by the technique).

We refer to this accumulated difference between real and virtual positions ( $\mathbf{P}_v(t1) - \mathbf{P}_r(t1)$ ) as *Translational Gain* (*Gain*, for short), and joined with Eq (6.2) is defined as:

$$\mathbf{G}(t1) = \sum_{t=0}^{t1} (\mathbf{k}(t) - (1,1,1)) \odot \Delta \mathbf{P}_r(t) \quad (6.3)$$

This, *Gain* implicitly represents the “extra displacement” that the technique has allowed between  $t=0$  and  $t=t1$ . Most techniques extending navigable space (i.e., main focus of this chapter) will aim for a *scaling policy* allowing high *Gain*.



**Figure 6.3** Drifted trajectory: A user moves along a closed path in the VE (blue line), but does not return to the same point in reality (red line). This offset is called *Drift*, and it changes with time (e.g., it will double if the user repeats the same virtual path again, dotted lines).

### 6.3.3 Defining *Drift*: Uncontrolled Gain along Closed Paths

Gain is also the source of the *Drift* problem. As illustrated in **Figure 6. 3**, when a user travels along a closed path in  $\mathbf{V}$  (i.e., blue path), it is most common that the user will not return to the same starting position in  $\mathbf{R}$  (i.e., end of red line). In the example, if the user was to repeat the path twice (i.e., blue path), he/she would have drifted towards the limits of the real space available (i.e., end of dotted red line). However, he/she would believe to be back at the same point (the headset shows he/she is right at the origin of  $\mathbf{V}$ ). Parts of the VE that were accessible before will no longer be available (e.g., our user cannot complete the path a third time), and the initial mapping between spaces  $\mathbf{R}$  and  $\mathbf{V}$  (*navigable space*) has been compromised. We draw inspiration from the exploration in [Chapter 5](#) to model this effect, describing *Drift* as “the *Gain* accumulated along a closed path in  $\mathbf{V}$ ”, being only defined if  $P_V(t1)=P_V(0)$ :

$$D(t1) = G(t1) \quad \Leftrightarrow \quad P_V(t1) = P_V(0) \quad (6.4)$$

This allows us to formalize the goal of our *Drift* correction techniques. The techniques should allow for high *Gain* (extended navigation), while ensuring *Drift* remains small (maintaining the original alignment between  $\mathbf{R}$  and  $\mathbf{V}$ ). We also focus our exploration on cases where *Drift* effects are most detrimental, particularly medium-size VEs (i.e., dozens of square meters) using reduced real spaces (i.e., 2.5x2.5m).

### 6.3.4 Estimated *Drift*: Approximation for Open Paths

As discussed in [Chapter 5](#), *Drift* correction methods will require an estimation of the current *Drift* at any point in time. However, *Drift* is only defined for closed paths (Eq (6.4)). A simple approach is to estimate the *Drift* that would be present if the user wished to return to the starting position ( $P_V(0)$ ) from the current position ( $P_V(t1)$ ). We denote such estimated *Drift* as  $\bar{D}(t1): \mathbb{R} \rightarrow \mathbb{R}^3$ , and compute it as a two-step process, described in the following subsections: **i)** simulate a path returning to the origin; and **ii)** estimate *Drift* based on current state (i.e., position, velocity) and the simulated path.

#### 6.3.4.1 *Simulating Path Returning to Centre:*

We use a very simple model, simulating that the user goes back to the origin following a straight line. To approximate user speed (i.e., required by some techniques ([Interrante et al., 2007](#))), we assume a conventional human motor control scheme ([Kawato, 1999](#); [Hayhoe et al., 2009](#)), where users initially collect information from the environment (plan

path), and then follows a ballistic locomotion stage (Horn et al., 1990), initially accelerating towards the target and slowing down upon arrival.

Let  $L$  and  $l(t)$  be the initial and current distance from the user to the centre respectively,  $v_0$  the user's initial speed ( $v_0 = \|\Delta \mathbf{P}_R(t_1)/\Delta t_1\|$ ) and  $v_M$  the maximum walking speed (Argelaguet, 2014; Zank et al., 2015). User speed is then approximated as in Eq (6.5):

$$\left\| \frac{\Delta \mathbf{P}_R(t)}{\Delta t} \right\| = \begin{cases} v_0 - 2(v_M - v_0) \cdot l(t)/L, & l(t) < L/2 \\ v_M - 2(v_M - v_0) \cdot (l(t) - L/2)/L, & l(t) \geq L/2 \end{cases} \quad (6.5)$$

Once the speed ( $\|\Delta \mathbf{P}_R(t)/\Delta t\|$ ) is determined, the user displacement is computed easily, by multiplying this magnitude by the unitary vector in the direction of the origin. This allows us to simulate both the user's real position and speed along its path back to the centre of the VE.

#### 6.3.4.2 Estimating Drift:

In order to estimate the *Drift*, we need to consider the current location and the simulated path (described above). Let's assume that at the current time  $t_1$ , the user is at positions  $\mathbf{P}_R(t_1)$  and  $\mathbf{P}_V(t_1)$ , with  $\mathbf{G}(t_1)$  being the current gain accumulated (as per Eq (6.3)). Also, let  $S: (t_1, t_2] \rightarrow \mathbb{R}^3$  be the simulated path returning to the centre computed as above.  $S(t)$  will describe the user's real position at each point in time ( $t \in (t_1, t_2]$ ).

*Estimated Drift* is simply computed by completing the closed path, that is, simulating Eq (6.1) along the path described by  $S$ :

$$\bar{\mathbf{D}}(t_1) = \mathbf{G}(t_1) + \sum_{t=t_1}^{t_2} (\mathbf{k}(t) - (1,1,1)) \odot \Delta \mathbf{S}(t) \quad (6.6)$$

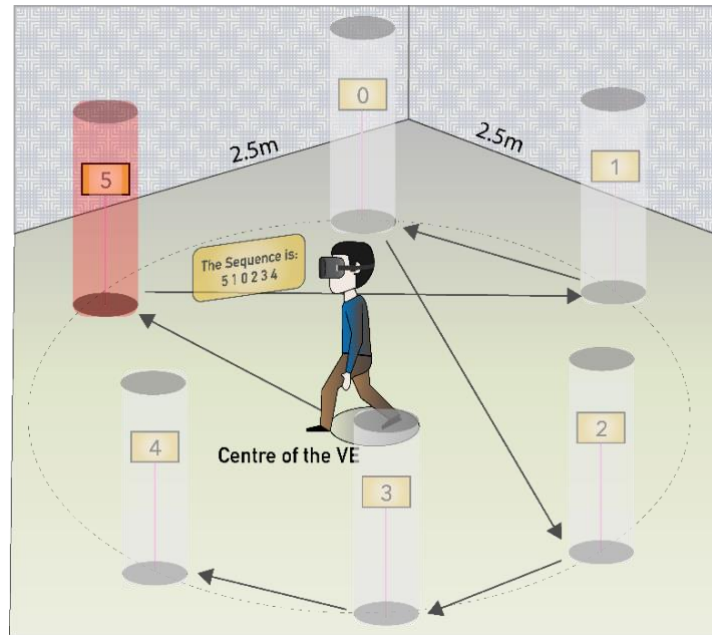
### 6.4 Study 1. Quantifying and Understanding *Drift*

As a preliminary step to guide our research, we wanted to gain insight on the magnitude of the *Drift* effect, to better assess its potential harmful effects on the usability of *scale-adaptive* techniques. Particularly, we used the *NaviFields* technique proposed in Chapter 5 with a navigable space of 168m<sup>2</sup>. We did so by fixing the maximum scaling factor to  $M=8$  and replicating the testbed environment used in Chapter 5 (i.e., six flag targets, placed in a hexagonal pattern, 8m away from the origin of  $\mathbf{V}$ ; inner radius of relevance areas 0.25m; outer radius 2.75m; all feedback cues replicated). The experiment was performed in an empty room-size tracking space of 2.5x2.5m.

Participants were asked to complete one training task and six navigation tasks. Each task required users to travel along sequences of 6 flags (L6) or 12 flags (L12) (see **Figure 6.4**).



Participants started the task from the centre and the “next flag” was always highlighted (red cylinder in **Figure 6.4**). For the training task, natural 1:1 navigation was used, allowing users to get used to the task and feedback. The six actual tasks using *NaviFields* followed, with each path length (L6 or L12) being repeated three times. Although each path was randomly generated, we ensured that all travel distances were the same, for paths of the same length (e.g., all paths of 6 flags were 54.4m long). The study was approved by the local ethics committee and we collected data from 12 participants (5 females, mean age=28.38 years old,  $SD=4.62$ ).



**Figure 6.4 Travelling task: Participants travelled a sequence of target flags (L6 and L12) in a shown sequence, starting from the centre of the VE.**

#### 6.4.1 Analysis of *Drift* Effects

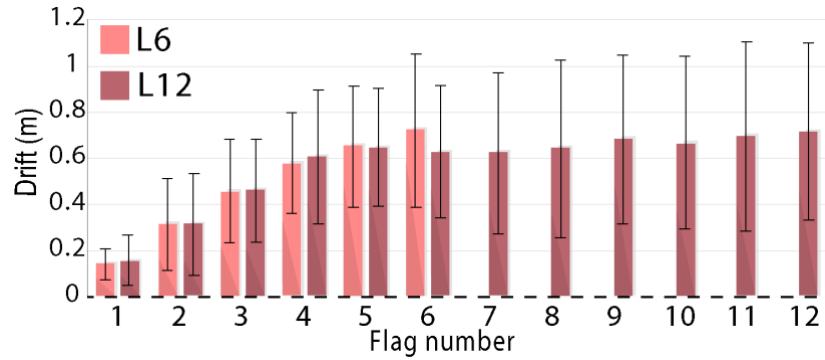
**Figure 6.5** shows the evolution of the *estimated Drift* (average magnitude and  $SD$ ), as our participants progressed along their flags. This average magnitude increased almost linearly during the first steps of the task (flags 1-6) and seemed to settle above 0.6m afterwards. This is a very high value for a 2.5x2.5m tracking area, practically reducing the usable space to 1.3x1.3m. Actually, 4 out of 12 participants could not complete their tasks because of the *Drift* (target areas not reachable), forcing us to enrol new participants to complete our sample. Thus, this is an optimistic estimate, as results are from successful participants, where *Drift* was not so high.

We computed the average *estimated Drift* magnitude at each flag for each participant (average across their three repetitions). An ANOVA analysis showed significant interactions between *estimated Drift* and flag number for both our 6-flag paths ( $p<0.001$ ,



$\eta^2=0.807$ ) and 12-flag paths ( $p<0.001$ ,  $\eta^2=0.830$ ), indicating that *Drift* effects increase, the more the user navigates. This interaction is not present ( $p=0.499$ ,  $\eta^2=0.126$ ) for the 6 last flags of our longer paths (i.e., flags 7-12), and hence we cannot disregard the hypothesis that *Drift* settles after the initial 6-flag stage.

A second observation worth noting is that the variance of the *estimated Drift* was very high. This could indicate that the magnitude of the *Drift* might be more difficult to predict as the length of the path increases (i.e., the longer the user is navigating). This is understandable, as *Drift* will evolve according to user's decisions (i.e., where she/he wants to go), becoming less predictable the longer that the user is navigating. However, this trend is not so obvious towards the end of the path (i.e., variance is more constant, between flags 7-12). Our ANOVA tests showed no significant interactions between the number of flags and the variance of the *Drift* (for any path lengths), so we cannot confirm these observations.



**Figure 6.5** Evolution of the average of estimated *Drift* for each path length (L6 and L12) and flag (1-12). Error bars represent SD. The bottom dashed line represents no *Drift*.

## 6.5 *Drift* Correction Techniques

The previous evaluation illustrated the very detrimental effects that *Drift* might have for the general use of *scale-adaptive* techniques. Four of sixteen users did not complete longer paths, and the *Drift* magnitudes observed ( $0.7 + 0.4$  m) were large, especially for such a small tracking space. In this section, we describe techniques to correct this effect. We first provide a simple general model for *Drift* correcting techniques. We then describe two approaches: *Derivative correction*, based in our initial model presented in [Chapter 5](#); and *Angular correction*, based on the relationship between the user's displacements and the *Drift* vector.

### 6.5.1 A Generic Model for *Drift* Correction Techniques

As introduced earlier, our techniques should allow for high *Gain* (i.e., navigate medium/large VEs from reduced real spaces), while ensuring *Drift* remains small (i.e., maintain initial mapping between  $\mathbf{R}$  and  $\mathbf{V}$ ). Also, we wished to maintain the initial properties of the navigation technique: **i)** displacements must retain the direction of the user's real motion (i.e., translational gain); and **ii)** correction must add minimal changes to the normal behaviour of the technique.

Based on these constraints, our correction techniques are a simple modification of the general case in Eq (6.1):

$$\mathbf{P}_V(t1) = \mathbf{P}_V(t1 - \Delta t) + c(t1) \cdot \mathbf{k}(t1) \odot (\Delta \mathbf{P}_R(t1)) \quad (6.7)$$

Here  $c: \mathbb{R} \rightarrow [1-\varepsilon, 1+\varepsilon] \subset \mathbb{R}$  describes a correction (scalar) function, which simply modulates the magnitude of the scaling policy (i.e., does not affect its direction). Besides, the function is bounded to a small range around 1 (i.e.,  $\varepsilon$  should be close to 0), to ensure that the behaviour of the corrected technique is mostly driven by the technique's scaling policy (i.e., the corrected behaviour remains similar to that of the uncorrected technique).

### 6.5.2 Derivative Correction

Our first technique is directly derived from the *Drift* model described in [Chapter 5](#), with the correction being driven by the variation in *Drift* at each point in time (see Eq (6.8)). For instance, if current displacements are increasing *estimated Drift* (i.e., positive derivative of  $\bar{\mathbf{D}}(t)$ ), displacements will be scaled up, to minimize *Drift*-increasing displacements:

$$c_{Deriv}(t) = \begin{cases} 1 - \varepsilon & , \alpha \cdot \Delta \|\bar{\mathbf{D}}(t)\| / \Delta t < -\varepsilon \\ 1 - \alpha \cdot \Delta \|\bar{\mathbf{D}}(t)\| / \Delta t & \\ 1 + \varepsilon & , \alpha \cdot \Delta \|\bar{\mathbf{D}}(t)\| / \Delta t > \varepsilon \end{cases} \quad (6.8)$$

We use two parameters to fully define this function. Parameter  $\varepsilon$  (mentioned above) defines the *bounds* of the correction (i.e., how much the corrected scaling factors can deviate from that of the basic technique). Parameter  $\alpha$  defines the *aggressiveness* of the correction (how quickly the technique reacts to derivative changes to try to correct *Drift*).

### 6.5.3 Angular Correction

The second technique looks at the correlation between the current displacement and the *Drift*. This is based in the implicit relationships between *Drift*, *Gain* and  $\Delta \mathbf{P}_R(t)$ . Given the behaviour of *scale-adaptive* techniques (translational gain), *Gain* increases/decreases

in the direction of  $\Delta \mathbf{P}_R(t)$ ). Given that *Drift* is a specific type of *Gain* (along a closed path), it will also evolve in the direction of  $\Delta \mathbf{P}_R(t)$ .

This second technique models this insight. If a displacement is aligned to the direction of the current *estimated Drift*, it will take this displacement as an opportunity to correct *Drift* (increase/decrease scaling). However, displacements perpendicular to the *Drift* are ignored, as they cannot help correcting the *Drift* effect. This is modelled by Eq (6.9), where  $\mathbf{u}_{\bar{\mathbf{D}}}(t)$  and  $\mathbf{u}_R(t)$  denote the unitary vectors in the direction of  $\bar{\mathbf{D}}(t)$  and  $\Delta \mathbf{P}_R(t)$ , respectively (and dot product being used to determine their relative alignment). The *Angular* technique defines the same tuning parameters as the previous technique (i.e., *bounds* ( $\epsilon$ ) and *aggressiveness* ( $\alpha$ )).

$$\mathbf{c}_{Ang}(t) = \begin{cases} 1 - \epsilon & , \alpha \mathbf{u}_{\bar{\mathbf{D}}}(t) \mathbf{u}_R(t) < -\epsilon \\ 1 - \alpha \mathbf{u}_{\bar{\mathbf{D}}}(t) \mathbf{u}_R(t) & \\ 1 + \epsilon & , \alpha \mathbf{u}_{\bar{\mathbf{D}}}(t) \mathbf{u}_R(t) > \epsilon \end{cases} \quad (6.9)$$

## 6.6 Testing & Tuning our Correction Techniques

Our first study revealed that *Drift* could have very detrimental effects on the continued usage of *scale-adaptive* techniques, justifying the need for correction algorithms. Our analysis also revealed a high variability in the magnitude of this effect, reflecting the fact that in the end, it is the user who is in control, and *Drift* will be determined by the users' decisions (i.e., where/when they want to go). This variability poses a serious challenge when trying to assess the best configuration for our correction techniques.

On one hand, values for parameters  $\epsilon$  and  $\alpha$  should be kept low. Particularly, a low *bounds* value ( $\epsilon$ ) is needed so that the technique's behaviour is mostly driven by its own *scaling policy* (and not by our attempts to correct *Drift*). Low *aggressiveness* ( $\alpha$ ) is required to prevent corrections from reacting too quickly to small changes in  $\bar{\mathbf{D}}(t)$ , as this could introduce jitter in the scaling factor, even if the user is moving smoothly. On the other hand, both  $\epsilon$  and  $\alpha$  need to be high enough as to maintain a reduced amount of *Drift* and allow for a continued usage of the navigation technique.

This section explores how different configurations of  $\epsilon$  and  $\alpha$  affect the behaviour of our correction techniques (*Derivative* and *Angular*) in terms of *Drift*. We first describe how we made use of the data recorded during Study 1 to simulate the effects of a given configuration (i.e., correction technique and specific values of  $\epsilon$  and  $\alpha$ ). We then describe the configurations simulated and the results obtained.

### 6.6.1 Data Driven Simulation

We reused the paths recorded during our first study (72 paths of 54.4m and 108.8m), simulating a user following the recorded paths with different configurations (i.e., correction techniques and values for  $\epsilon$  and  $\alpha$ ), and computing the resulting  $\bar{D}(t)$ .

We could not use the users' real displacements (i.e.,  $P_R(t)$ ) to guide our simulations. By changing the configuration (technique and parameters), these displacements could result in very different paths in VR, not necessarily going through the same points/flags in the VE than the user we recorded.

Instead, our simulations are guided by the users' virtual paths (i.e.,  $P_V(t)$ ). That is, given a specific configuration (i.e., technique, parameters), we compute the real displacements that the user should have made to follow the virtual path recorded. This implies the assumption that users are in control of the navigation (they know the path they want to follow and plan their physical displacements to match it).

### 6.6.2 Simulation of Correction Configurations and Results

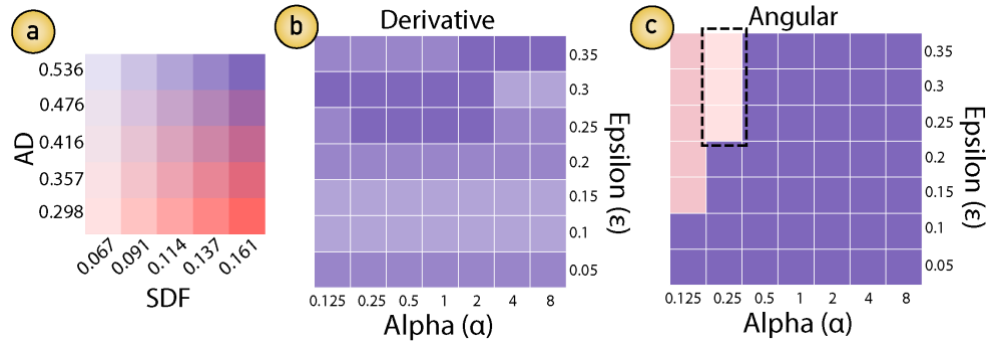
We performed a brute force search of our solution space for both techniques (*Derivative* and *Angular*), using increasing (but low) values of *bounds* ( $\epsilon \in \{0.35, 0.3, 0.25, 0.20, 0.15, 0.10, 0.05\}$ ). The aggressiveness factor needs to account for the difference in units between the correction criteria (e.g.,  $\Delta\|\bar{D}(t)\|/\Delta t$ ), and the correction range (e.g.,  $[1-\epsilon, 1+\epsilon]$ ), and was hence explored following a geometric series ( $\alpha \in \{0.125, 0.25, 0.5, 1, 2, 4, 8\}$ ), each covering a different order of magnitude. This resulted in 49 different tests for each correction technique (*Derivative* and *Angular*) resulting in 98 configurations in total, aiding us in the search of the best configuration for each technique, and providing insight on their general behaviour.

**Figure 6.6** provides an overview of the results obtained from this search, showing the final Average (AD) and Standard Deviation (SDF) of *estimated Drift* for each configuration. The colour code in our Bi-variate chart is scaled (see **Figure 6.6(a)**). As minimum values for AD and SDF (pink colour), we use the minimum values found across all 98 configurations. As maximum values (purple colour), we use the AD and SDF of the uncorrected technique (i.e., any value higher than the uncorrected technique is considered unacceptable).

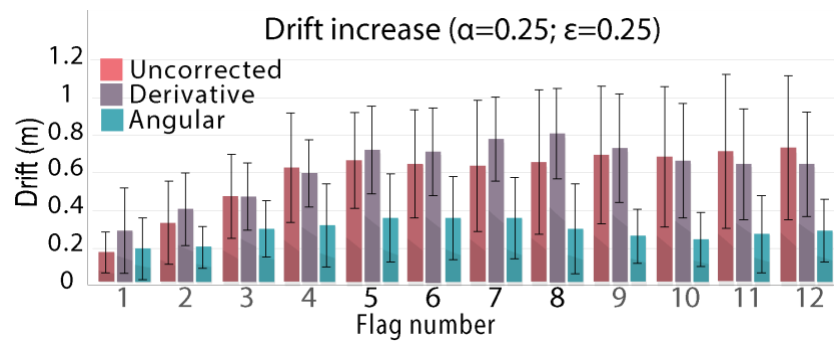
The *Derivative* technique showed very poor results, always with high  $\bar{D}(t)$  (purple colours, in **Figure 6.6(b)**). Paired comparisons with Bonferroni corrections were run,

comparing the *estimated Drift* of the uncorrected technique with each configuration of *Derivative* tested. No significant differences were found in most cases and, where differences were found, *Derivative* always performed worse than the basic uncorrected technique indicating that the approach proposed in [Chapter 5](#) is not an effective approach to correct *Drift*. Results for paired tests for all *Derivative* and *Angular* tests are available in our [Appendix 3](#).

The *Angular* technique did show very good results for some configurations (pale pink in top-left of **Figure 6.6(c)**). Our results show that the *Angular* technique reduces *Drift* for higher values of  $\epsilon$  and low values of alpha  $\alpha$  (best result for  $\epsilon=0.25$  and  $\alpha=0.25$ ). The first result (good correction for higher  $\epsilon$  values) is to be expected, as higher *bounds* ( $\epsilon$  values) allow *Angular* to apply larger corrections to the *scaling policy*.



**Figure 6.6 Overview of the results (data driven simulation):** (a) Bi-variate map, showing colour code used for *Average  $\bar{D}$*  (AD) and *Standard Deviation of  $\bar{D}$*  (SDF); (b) The *Derivative* approach showed very poor results (not improving the behaviour of the uncorrected technique). (c) The *Angular* technique showed a clear sweet spot (pink area in top-left), for low values of  $\alpha$  and mid-high values of  $\epsilon$ .



**Figure 6.7 Comparison of *Drift* ( $\alpha=0.25$ ,  $\epsilon=0.25$ , best configuration) for our techniques, showing evolution of  $\bar{D}$  as users travelled across flags 1-12.**

The explanation of the second result (good correction for  $\alpha=0.25$  or  $\alpha=0.125$ ) required a more detailed understanding of the behaviour of the *Angular* technique. The technique (Eq (6.9)) uses the alignment between the *Drift* vector and the user displacement to guide correction. More specifically,  $\mathbf{u}_{\bar{D}}(t) \mathbf{u}_R(t)$  describes the cosine of the angle between

these two directions. This provides a value in the range  $[-1,1]$ , which is then scaled by the *aggressiveness* to the range  $[-\alpha, \alpha]$ , with the final correction capped by  $\epsilon$  (range  $[-\epsilon, \epsilon]$ ).

For our best case ( $\epsilon=0.25$  and  $\alpha=0.25$ ), the *aggressiveness* naturally maps the  $[-1,1]$  range to the range  $[-0.25, 0.25]$  allowed by  $\epsilon$ . Displacements in the direction of *Drift* (0 degrees) will result in a maximum correction (0.25), which will decrease to zero as the angle reaches 90 degrees.

However, a higher value of  $\alpha$  will make the *Angular* technique less selective to directions (*Drift* to displacement angles). For instance, for ( $\epsilon=0.25$  and  $\alpha=0.5$ ), a displacement angle of 60 degrees respective to  $\bar{D}(t)$  will already provide a maximum final correction value of  $0.5 \cdot \cos(60)=0.25$ . A smaller angle (e.g., 30 degrees) will not provide more correction (e.g.,  $0.5 \cdot \cos(30)=0.43$  is bigger than  $\epsilon=0.25$ , so correction will be capped to 0.25). Thus, techniques with  $\alpha > \epsilon$  will scale large angles (i.e., poor alignment) just as much as small angles (i.e., good alignment), making the correction technique unreliable. Actually, all configurations with  $\alpha > \epsilon$  failed to correct *Drift* and in some cases resulted in very adverse effects e.g.,  $\epsilon=0.25$ ;  $\alpha=8$  resulted in a final AD of 2.37m (more examples are presented in [Appendix 3, Figure S1](#)).

Similarly, values of  $\alpha$  smaller than  $\epsilon$  do not make good use of the correction range allowed by  $\epsilon$ . For  $\alpha=0.125$ , the range  $[-1,1]$  will always be mapped to  $[-0.125, 0.125]$ , even if  $\epsilon$  would allow for higher corrections (e.g.,  $\epsilon = 0.3$ ). Actually, paired analysis comparing configuration ( $\epsilon=0.125$  and  $\alpha=0.125$ ) with other configurations (see [Appendix 3, Table S1](#)) with higher  $\epsilon$  never show any significant differences/improvements.

Our results and analysis clearly show that the *Angular* technique with  $\epsilon=\alpha$ , will result in an optimum use of its correction range. Based on this, we selected  $\epsilon=0.25$  and  $\alpha=0.25$  for our following studies (*Drift* results in [Figure 6.7](#)).

## 6.7 Study 2. Generability and Robustness of our Correction Techniques

Our previous step allowed us to assess the effectiveness of our correction technique for a specific technique and set of paths, allowing us to recommend a specific configuration (*Angular* technique,  $\epsilon=0.25$ ,  $\alpha=0.25$ ). In this study we tested if our correction would still be effective for other techniques and environments but, more importantly, with real users.

Particularly we tested two techniques: *Seven League Boots* ([Interrante et al., 2007](#)) and *NaviFields* (proposed in [Chapter 5](#)), as representatives of two different paradigms. In

Seven League Boots, scaling is exclusively dependent on the user's speed, being a good representative of an *ego-centric* technique (de Haan et al., 2009). In *NaviFields*, the scaling factor depends exclusively on the location of the user in the VE (i.e., how relevant that point is), being an example of an *allocentric scale-adaptive* technique. Besides reducing *Drift* for these techniques, we also wanted to test if our *Drift* correction techniques introduced any ill effects to the navigation experience. That is, if either the user performance or subjective impressions (e.g., sense of presence, sense of control) were distorted as a result of our dynamic (maybe unexpected) corrections. It must be noted that our intention is not to compare the performance of *NaviFields* vs *Seven League boots*, but to assess if our correction can address *Drift* in both cases and without affecting their overall behaviour.

### 6.7.1 Seven League Boots Implementation (7LB)

*Seven League Boots* (7LB) was implemented as a particular case of our corrected *scale-adaptive* techniques (i.e., using Eq (6.7)), with its *scaling policy* only determined by user's real speed ( $\mathbf{v}_R = \Delta \mathbf{P}_R(t1)/\Delta t$ ), and defined as follows:

$$\mathbf{k}(t) = \begin{cases} 1 & , \mathbf{v}_R < 0.2 \text{ m/s} \\ e^{1.62 (\mathbf{v}_R - 0.2)} & , 0.2 < \mathbf{v}_R < 1.4 \text{ m/s} \end{cases} \quad (6.10)$$

This implementation is directly inspired by the original technique. First, the range of speeds chosen (e.g.,  $\mathbf{v}_R \in [0, 1.4)$  m/s) corresponds to the walking speed of an average adult. The exponent selected (i.e., 1.62) allows scaling of up to 7 at maximum walking speed, like the original technique.

### 6.7.2 Environment and NaviFields Implementation (NF)

Being an *allocentric* technique, the explanation of the *NaviFields* (NF) implementation used is tied to the VE (and *navigation field*) used, and they are here described together.

First, we used a smaller environment than in Study 1. This was partially a consequence of the implementation of 7LB. Assuming all walking speeds were equally probable, the average scaling provided by 7LB was 2.79 (i.e., integral of  $\mathbf{k}(t)$  divided by 1.4). The environment was scaled according to this factor, with flags located at 2.79 m from the centre, and following the same hexagonal distribution as in Study 1. Although we do not compare 7LB vs NF, we wanted to test the *NF* technique in an environment different than the one used in Study 1 (i.e., test if the configuration tuned for a scaling factor of 8 was still valid in other environments). Thus, we decided to test NF using the same

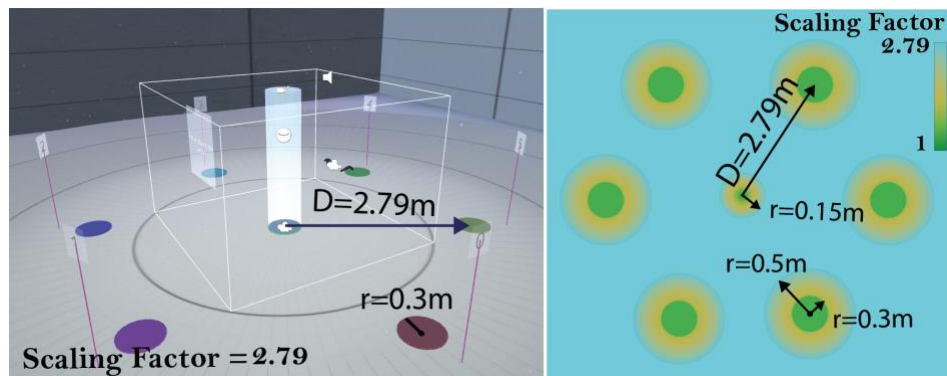
environment that we use for 7LB, modifying the *navigation field* according to its new size. More specifically, we changed the outer radius of the 6 flags to 1.01m and further modified the *navigation field*, by adding an extra area of interest in the centre of the room (inner radius 0.15m, outer radius 1.01m). The final VE resulting and its *navigation field* used are shown in **Figure 6.8**.

### 6.7.3 Navigation Tasks

We used a travelling task similar to that in Study 1, but with 42 flags in each path. The intention was to test if our correction techniques increased the amount of time users could use the technique, before the *Drift* forced them to stop (i.e., next flag not reachable from the tracking space). Thus, 42 was the maximum length that users could navigate, but we expected most users would not be able to finish the task. Each 42 flags sequence was internally structured as six repetitions of a 7-flag pattern. Each pattern started with the user navigating along 6 flags on the edges of the environment (*accumulation stage*) and in the seventh step of the sequence, the user was asked to return to the centre of the VE.

Returning to the centre was required to obtain a true measurement of the *Drift*. That is, our correction techniques use *estimated Drift* ( $\bar{D}(t)$ ) to apply corrections, but the real *Drift* effect ( $D(t)$ ) is only defined for closed paths (Eq(6.5)). Our design of the task (six 7-flag patterns) allowed for up to six measurements of *Drift* during the task and *Drift* could continue to accumulate from one 7-flag pattern to the next.

We used 6 flags for the *accumulation stage* of the patterns because, *Drift* did not seem to increase significantly with the number of flags after the 6<sup>th</sup> flag (see interactions between path length and *Drift* in Study 1). The order of the flags for the *accumulation stages* were pseudo-randomized, with equal probabilities for all flags but ensuring that the final path length was always constant (141.68m in the VE).



**Figure 6.8** Test environment and *navigation field* used for the second study, after modifying scale and adding extra regions.



#### 6.7.4 Experimental Design and Variables Measured

We conducted two independent studies, each one testing our correction method for one navigation technique (i.e., NaviFields (*NF*) or Seven League Boots (*7LB*)) and comparing the *Drift* effect with and without our *Angular* correction ( $\epsilon=0.25$ ;  $\alpha=0.25$ ). In each study, we compared two conditions: *NF* (original technique) and *NFC* (corrected) for the first study, whilst *7LB* (original technique) and *7LBC* (corrected) for the second study. Participants completed three repetitions of our Navigation task for each condition (*NF*, *NFC*) and (*7LB*, *7LBC*). Both studies were approved by our ethics committee and we collected data from 12 participants (2 females, mean age=31.39y, SD=5.33) in the first study and 12 participants (4 females, mean age=28.38 years old, SD=4.90) in the second study. Our software automatically recorded several dependent variables during both studies (acronyms used summarized in **Table 6.1**). We conducted separate statistical analyses for each study. However, results from both studies are presented together, for brevity and due to their similarity.

Acronym	Description
<i>D7, D14, D21, D28, D35, D42</i>	<i>Drift</i> at the end of each 7-flag pattern
<i>FR</i>	Number of flags reached
<i>TPF</i>	Time required to reach each flag
<i>RDF</i>	Real distance per flag

**Table 6.1 Variables measured to assess technique performance.**

##### 6.7.4.1 Drift Correction Variables

The first group of variables tested the performance of our correction technique. As described above, we measured the *Drift* at the end of each 7-flag pattern (identified as *D7, D14, D21, D28, D35* and *D42*). We also measured the number of flags the user reached (*FR*) before *Drifting* out of the tracking space, assessing support for longer/sustained navigation.

##### 6.7.4.2 Travelling Performance and User Experience Variables

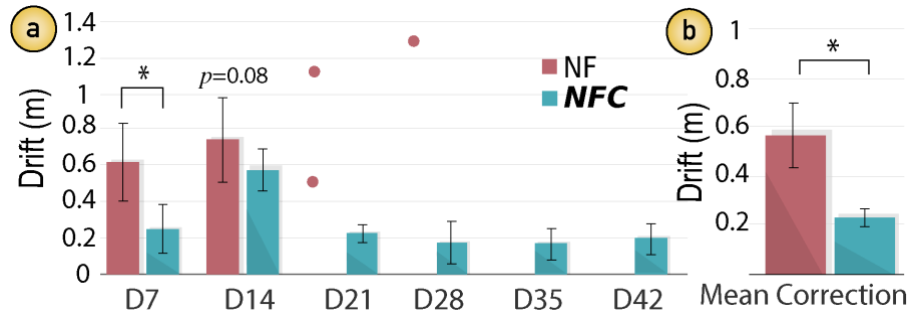
We measured the performance of each technique vs their corrected counterparts, to test our assumption that our corrections would not influence/hinder the usage of the technique. We specifically measured: time per flag (*TPF*) and real distance per flag (*RDF*). *TPF* measured the average time required to reach each flag (time navigated divided by *FR*) and *RDF* measured the average real distance that users moved to reach each flag (total real distance moved divided by *FR*). Please note that these corrections (dividing per *FR*) were necessary as the length of the trials could vary. We also wanted

to ensure that the corrections would neither affect/hinder the user's subjective perception of the techniques (when compared to uncorrected counterparts), due to any dynamic/unexpected adjustments in scale. To do so, we used 8 questions, with 5 of them selected from (Witmer et al., 1998) (questions 6,7,10,13&16, focusing on navigation). Specifically, we assessed: comfort, control, involvement, ease, naturalness, consistency, sense of moving and proficiency (see Appendix 3, Table S4, for a list of questions).

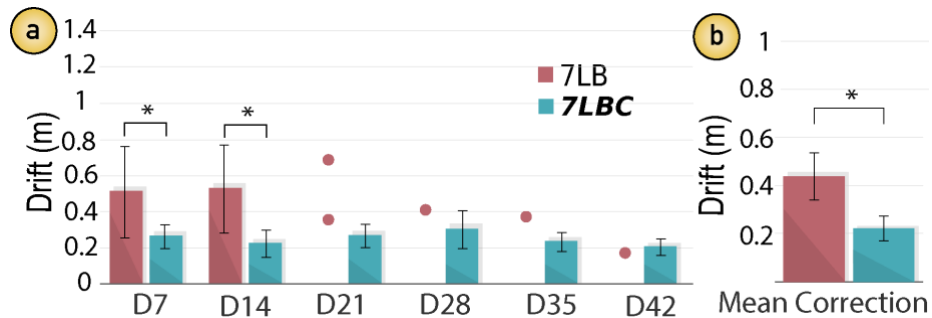
### 6.7.5 Analysis of Results

#### 6.7.5.1 Drift Correction Results

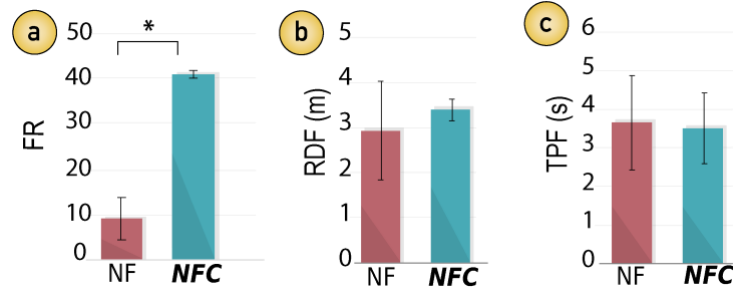
One-way Repeated Measures ANOVA was used, to test for differences on the real *Drift* between each technique and its corrected counterpart (i.e., *NF* vs *NFC*; *7LB* vs *7LBC*). The evolution of *Drift* along the various 7-flag stages (*D7-D42*) in each technique is shown in **Figure 6.9(a)** and **Figure 6.10(a)**. In both cases, our analysis showed significant differences between uncorrected techniques and their corrected counterparts for the first stage of flags (generally until *D14*). This analysis was affected by the low number of participants reaching the last stages without correction. For *NF*, only 2 participants reached *D21*, and only one reached *D28* (see **Figure 6.9(a)**). In the case of *7LB*, only one participant completed *D42* (see **Figure 6.10(a)**).



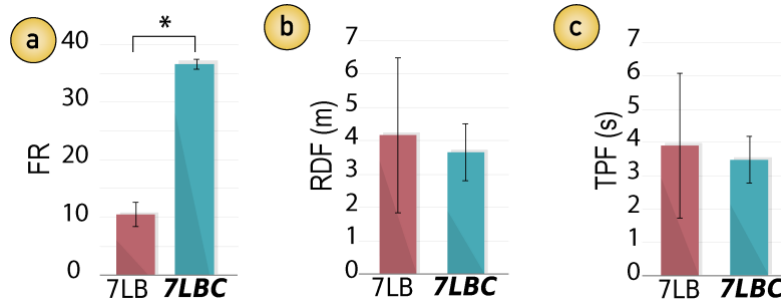
**Figure 6.9 Drift in NaviFields: Comparison of uncorrected (NF) and corrected (NFC) Drift. (a) Evolution of Drift along the various 7-flag stages (D7-D42). (b) Overall Drift along a path of 42 flags. Error bars represent SD.  $*=p<0.05$ .**



**Figure 6.10 Drift in Seven League Boots: Comparison of uncorrected (7LB) and corrected (7LBC) Drift. (a) Evolution of Drift along the various 7-flag stages (D7-D42). (b) Overall Drift along a path of 42 flags. Error bars represent SD.  $*=p<0.05$ .**



**Figure 6.11 NaviFields: Comparison of performance (NF Vs NFC) regarding (a) Number of flags reached, (b) Real distance per flag and (c) Time per flag. Error bars represent SD. \*= $p < 0.05$ .**



**Figure 6.12 Seven League Boots: Comparison of performance (7LB Vs 7LBC) regarding (a) Number of flags reached, (b) Real distance per flag and (c) Time per flag. Error bars represent SD. \*= $p < 0.05$ .**

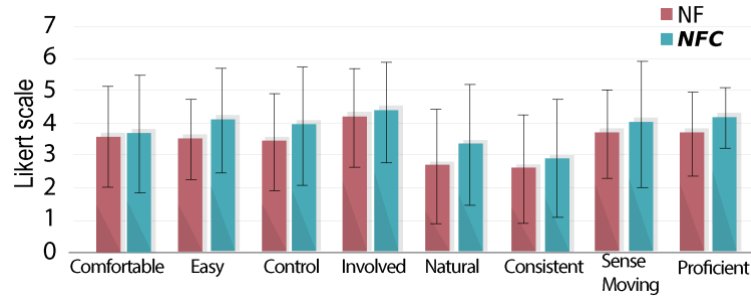
When all *Drift* measurements were considered together (D7-D42), our ANOVA tests revealed a significant reduction of *Drift* for *NFC* ( $F_{(1,11)}=59.6$ ,  $p < 0.001$ ,  $\eta^2=0.844$ ) as shown in **Figure 6.9(b)**, and for *7LBC* ( $F_{(1,11)}=79.74$ ,  $p < 0.001$ ,  $\eta^2=0.879$ ) as shown in **Figure 6.10(b)**. Besides showing that our *Angular* correction reduced *Drift* in both cases, the results show good agreement with the *estimated Drift* expected from our data driven simulations (corrected *Drift*  $\sim 0.2$  m; uncorrected  $\sim 0.6$  m), supporting the reliability of our estimation method.

Participants could also reach more flags in both experiments, when *Angular* correction was applied. Specifically, One-way Repeated Measures ANOVA showed significant effect on the number of flags users could reach (*FR*) between *NF* vs *NFC* (**Figure 6.11(a)**;  $F_{(1,11)}=641.68$ ,  $p < 0.001$ ,  $\eta^2=0.983$ ), and also between *7LB* vs *7LBC* (**Figure 6.12(a)**;  $F_{(1,11)}=91.84$ ,  $p < 0.001$ ,  $\eta^2=0.893$ ). While very few participants could complete D21 with uncorrected techniques, almost all participants were able to reach all 42 flags using *NFC* (10 out of 12) or *7LBC* (11 out of 12). These results both highlight how *Drift* can seriously limit the applicability of *scale-adaptive* techniques and how our correction techniques can greatly extend the amount of time and distance that the users can navigate before *Drift* becomes an issue. Besides, the fact that *Drift* tends to stabilize  $\sim 0.2$  m after the initial

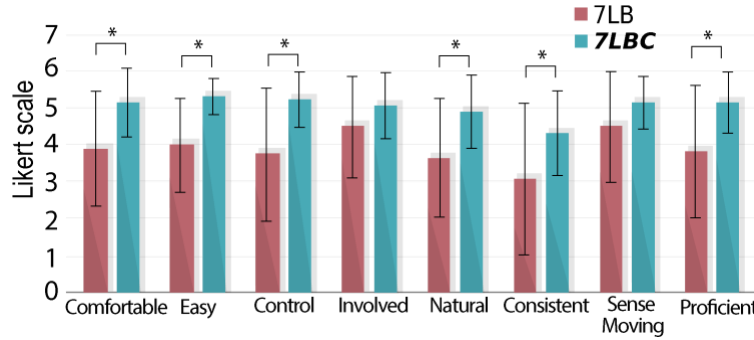
stages of the experiment (D21-D42) seems to indicate that the corrected techniques would allow sustained navigation.

#### 6.7.5.2 Effects of Correction on Performance and User Experience:

No significant differences were found between the technique (*NF* and *7LB*) and its corrected counterpart (*NFC* and *7LBC*) for time per flag (*TPF*) and real distance per flag (*RDF*). These results seem to confirm that our correction scheme (*Angular*,  $\varepsilon=0.25$ ,  $\alpha=0.25$ ) did not influence/hinder user's performance neither in terms of time or distance travelled (see **Figure 6.11(b-c)** and **Figure 6.12(b-c)**).



**Figure 6.13** Questionnaire results for *NF* and *NFC*.



**Figure 6.14** Questionnaire results for *7LB* and *7LBC*.

Also, the analysis of the questionnaire responses indicates that the correction techniques did not have negative effects on the users' experience. In the case of NaviFields, no significant differences could be found between *NF* and *NFC* for any of the 8 questions (**Figure 6.13**). Differences could be found in the case of 7 League Boots but, surprisingly, they revealed higher satisfaction with the corrected technique (see **Figure 6.14**). Significant differences between *7LB* and *7LBC* were found in terms of comfort ( $F_{(1,11)}=16.83$ ,  $p<0.01$ ,  $\eta^2=0.605$ ), ease ( $F_{(1,11)}=11.35$ ,  $p<0.01$ ,  $\eta^2=0.508$ ), control ( $F_{(1,11)}=6.907$ ,  $p<0.05$ ,  $\eta^2=0.386$ ), naturalness ( $F_{(1,11)}=9.27$ ,  $p<0.05$ ,  $\eta^2=0.457$ ), consistency ( $F_{(1,11)}=6.39$ ,  $p<0.05$ ,  $\eta^2=0.368$ ) and proficiency ( $F_{(1,11)}=7.65$ ,  $p<0.05$ ,  $\eta^2=0.410$ ).

We do not believe this is indicative of correction techniques improving experience but believe it could be a subjective bias. As the users managed to reach more flags with *7LBC* than *7LB* (before completing the questionnaire), this could give them a sense of achievement, which could translate to more positive subjective results. In any case, these results provide strong evidence that the correction technique does not add significant negative effects to the user experience.

## 6.8 Discussion

As introduced earlier, the size of the spaces available to end VR users influences the type of VR experiences and navigation techniques created by industry. While *scale-adaptive* stand as a feasible solution for this (i.e., use of physical displacements, addressing medium size VEs from relatively small real spaces), our study illustrates how unconstrained *Drift* can grow quickly (i.e., 4 out of 16 participants could not complete their 12-flag trials, while only 2 out of 12 reached flag number 21 in our second study), which would lead to users having to frequently interrupt their experiences (e.g., remove the headset and return to the centre of the tracking space), or can even result in collisions with objects placed around the designated tracking space.

Our experiments showed very encouraging results for the correction of such *Drift* effects, applied to two examples of such *scale-adaptive* techniques (an *ego-centric* and an *allocentric* one). *Drift* was consistently reduced but, more relevant than that, the inclusion of our technique greatly increased the amount of time and distance that our users could continue to navigate before *Drift* made targets unreachable (i.e., user drifting out of the tracking space). Most users (88%) could complete the whole test, and *Drift* tended to settle around stable magnitudes (~0.2 m), suggesting that sustained navigation over the medium sized VE used could be possible.

However, we believe that (corrected) *scale-adaptive* techniques will still require the use of other supporting techniques. Even if, as results suggest, sustained navigation could be achieved for a particular technique and environment, these techniques are still limited to medium size VEs, which would still be a limitation for the creation of commercial VR experiences (or VE experiences designed to be used in a reduced VR space). Also, even if *Drift* tends to remain stable over time, this still does not fully eliminate the possibility that the user drifts out of the real space (i.e., some of our test users did), and support techniques should be made available to be used in these cases. For instance, the use of redirected teleporting (Liu et al., 2018) can act as a great complement here,

simultaneously correcting *Drift* and also redirecting the user towards the centre of the VE. Using redirected walking technique can be also a suitable combination, however one of the main constraints could be the big space required from the technique to generate the motion redirection (by rotational and translational gains). Such solution (or combination of solutions) could enable VR experiences where navigation is driven by users' physical displacements most of the time, covering a variable size of the VE and better suited for current constraints around real space availability.

It is also worth noting that our correction technique is independent of the navigation technique used and can be applied to any technique following the model presented by Eq (6.1). The fact that our correction provided consistent results, even if applied to different techniques and VE (i.e., Study 1) is encouraging, as a potential solution applicable to other (or maybe any) *scale-adaptive* techniques, or even to other techniques from redirected walking making use of translational gain.

## 6.9 Conclusion

This chapter has presented an exploration of the technique-induced *Drift* effect, present in *scale-adaptive* techniques (i.e., VR navigation techniques involving physical displacements and translational gain). Such effect had received little attention, even though it affects a great range of VR navigation techniques and its effects (as seen in our study 1) can be very detrimental to the usage of these techniques, particularly for virtual environments deployed over reduced physical spaces.

We presented a general formulation for such *scale-adaptive* techniques compatible with a wide range of VR techniques. Such formulation allowed us to derive and formalize the factors that drive and limit their behaviour (*Gain*, *Drift*). We also proposed a simple but general model to correct this effect, being independent of the specific navigation technique.

Our studies have shown how it is possible to use this framework to model and correct *Drift* for both *ego-centric* and *allocentric* techniques, while not adding any significant changes to the basic behaviour of the technique. We believe our results could enable *scale-adaptive* techniques (or hybrid models making use of them) to become a very interesting alternative for current commercial VR experiences or experiences deployed in small real spaces, allowing for more extensive use of metaphors involving users' physical displacements.

The findings shown in this chapter not only highlight the relevance of *Drift* effects on navigation techniques but also highlight a potential of our solution in helping to reduce it, leveraging the use of techniques to extend the navigation space and extend the time of navigation in VR.

## Chapter 7

# Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments

In this chapter, we explore the third stage of interaction (ITas) of 3DUIs. We focused on the complexity of selection tasks due to occlusion when the VE is highly dense (e.g., point-clouds). According to our general approach we first, identify the ITa's requirements and then design a tailored ITe to meet their specific needs. We initially explore the benefits and constraints of mid-air and tangible interaction (under the frame touch-based interaction devices e.g., table-tops, tablets, smartphones, etc.). Then, we present a customized ITe that combines both 3D and 2D interactions to address the problem of occlusion (produced by scene density) and lack of stability during the task (produced by mid-air selection). We used a GPU-based solution to manage real-time computationally heavy interaction with ~1 million elements within the VE. The approach introduced in this chapter allows more natural interactions for selection tasks by combining 3D and 2D metaphors i.e., mimicking the way users interact in the real world (e.g., *pen-and-tablet* metaphor) involving haptic feedback common when using physical surfaces.

## 7.1 Introduction

Capturing the real-world using 3D scanning and imaging is becoming increasingly popular and widespread across a range of application domains. Examples include photometric scanning ([Izadi et al., 2011](#); [Cheveau, 2018](#)), medical image analysis (e.g., 3D MRI and CT scans ([Altahawi et al., 2018](#))), virtual exploration of infrastructure in VR/AR (e.g., Matterport and ARKit), and VR sculpting & painting ([Agrawala et al., 1995](#); [Prior, 2006](#)). Target selection plays a critical role for interacting with such



unstructured 3D contents for clean-ups, analysis, and editing. For instance, scientists analysing MRI scans often need to use slicing planes to inspect and select relevant areas (Hinckley et al., 1994; McKeown et al., 2003; Peeters et al., 2004; Janoos et al., 2009; Obermaier et al., 2015). However, such selection tasks are well known to be challenging due to the complexity of the dense environment (Yu et al., 2012). A typical 3D scanned model can contain around six million points. In such a dense space, occlusion is a major challenge for accomplishing selection tasks (LaViola Jr et al., 2017).

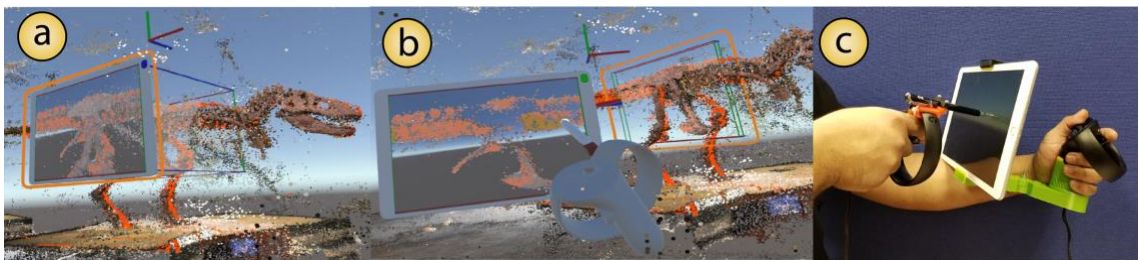
Virtual Reality (VR) is a promising platform for 3D data manipulation and analysis (Cruz-Neira et al., 1993; Ribarsky et al., 1994). A key aspect of VR is the high degrees of freedom (DOF) offered by its input modalities, such as 3D controllers (6-DoF (LaViola Jr et al., 2017)) and mid-air hand gestures (more than 25-DoF (Jones et al., 2006; Sridhar et al., 2015)). However, despite the freedom of mid-air interactions, these prevalent input modalities lack haptic feedback and stability required for precise interactions (Herndon et al., 1994; Argelaguet et al., 2013). Furthermore, in contrast to 2D interaction (e.g., touchscreens), fine motor control tasks (Keefe et al., 2007; Arora et al., 2017) are also physically-demanding in arbitrary 3D scales due to our ergonomic limitations (Arora et al., 2017).

Recent studies have proposed hybrid techniques that combine the benefits of both free 3D mid-air gestures and precise 2D tactile input (Arora et al., 2018; Dias et al., 2018) to aid VR interactions. For example, in SymbiosisSketch (Arora et al., 2018) having a 2D *pen-and-tablet* metaphor within a 3D world, helped in improving precision and ballistic actions by constraining the motion along the Z axis and providing haptic feedback (Lindeman, 1999; LaViola Jr et al., 2017). These hybrid approaches have been therefore used for navigation (Medeiros et al., 2013; Dias et al., 2018), 3D drawing (Arora et al., 2018) and simple selection (i.e., limited to a low number of targets) (Medeiros et al., 2013; Wang et al., 2015; Afonso et al., 2017). More recently, Surale et al. (Bhaskar Surale et al., 2019) contributed a comprehensive design space of solid modelling techniques enabled by a tracked 6-DOF tablet in VR. These explorations show promising values of hybrid techniques, particularly to improve the stability of the interaction. However, it is still unclear whether such proposed hybrid approaches could address the occlusion problem for selection tasks in highly dense Virtual Environments (VE's). Specifically, how do we design a hybrid technique that can help users comfortably and accurately

select a specific amount of points in a highly occluded environment? As well as, how can we evaluate such technique?

In this chapter, we propose a hybrid selection technique for dense point-clouds in VR. Our technique maps the complex 3D multi-target selection task into a hybrid workflow that combines both mid-air and tablet interactions (see **Figure 7.1**). This mapping was done through a variation of known slicing plane techniques (Hinckley et al., 1994), i.e., our approach consists of a “*Slicing-Volume*” that encloses desired 3D objects and project them onto an tablet view enabling 2D interactions for more precise selection. We implemented our technique in a standard VR system with two mid-air controllers. A touch pen was attached to the dominant-hand controller, and a multi-touch tablet to the non-dominant-hand controller (see **Figure 7.1(c)**). The user places the *Slicing-Volume* around the selection targets using her non-dominant hand. Then, selection and refinement can be carried out on the tablet display using her dominant hand i.e., a *pen-and-tablet* metaphor (see **Figure 7.1**).

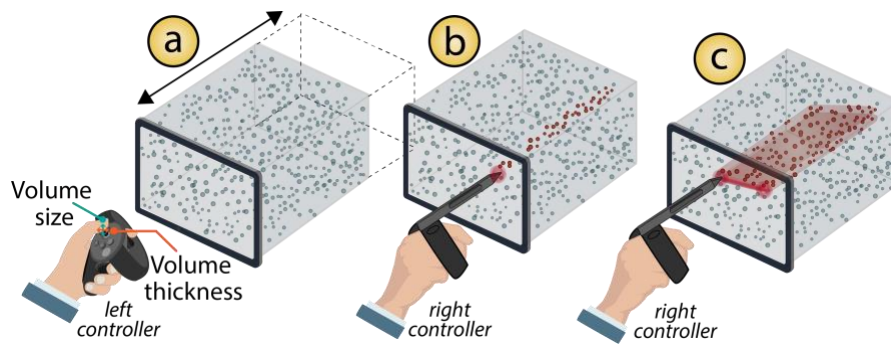
The mapping from the *Slicing-Volume* to the tablet essentially reduces the complexity of the selection task. The dimensions of the *Slicing-Volume* can be adjusted, which enables either selection of a large set of points (by increasing its size) or selection of occluded points (by decreasing its thickness). When a user touches on the tablet, our technique casts a ray from the touch point perpendicular to the tablet surface to select all points along this ray (which length is defined by the *Slicing-Volume* thickness) (see **Figure 7.2**). Thus, users can select 3D points using continuous tactile touch, which is physically more comfortable and precise (LaViola Jr et al., 2017).



**Figure 7.1 Slicing volume 3D-2D interaction:** The user places the volume (which is initially attached to a virtual tablet) on the desired area in VR (a), and adjusts the volume size and thickness to deeply explore the model, managing occlusion (b). The points enclosed within the volume are projected to a virtual tablet view. Then the user selects the visible points on the tablet surface using a pen-and-tablet metaphor. The selection on the tablet is then propagated along the Slicing-Volume thickness and updated in the 3D model. In the real world, the user holds a real tablet and a real pen attached to 3D controllers, which provide stability and haptic feedback for precise selection.

We conducted a user study to explore whether this hybrid technique improves accuracy in dense selection, and if so, which aspects of the system contribute to such improvement (i.e., the stability given by the physical tablet surface, the extra visualization given by the added tablet view, or a combination of both). To answer these questions, we compared three main conditions: (1) *Mid-air* only, (2) *Mid-air & Virtual Tablet* and (3) *Mid-air & Real Tablet* in two point-cloud models with different levels of selection difficulty, i.e., occlusion level (highly occluded and mildly occluded). Our results demonstrate that, compared to mid-air only interactions, our hybrid selection technique significantly improved the accuracy of selection tasks for highly dense environments. Our study indicates that this is primarily due to haptic feedback afforded by the physical tablet, rather than the extra visualization from virtual tablet mode.

In summary, unlike current hybrid techniques proposed in the literature (mostly limited to selection with a low number of targets, sparsely distributed across space), the main contributions of our work are: (a) we explore the potential of *tablet-in-VR* approaches to address the occlusion problem of VR selection tasks, (b) we propose and implement a real-time VR selection technique tailored for highly dense and highly occluded environments, (c) we conduct a user study to understand the factors influencing users' performance according to 3 independent factors: the use of extra visualizations (*Mid-air* vs *Virtual Tablet* modes); physical support (*Virtual* vs *Real Tablet* modes); and level of task complexity (highly occluded and mildly occluded 3D models).



**Figure 7.2 Propagation technique:** (a) The user increases/decreases the volume size and thickness using the left joystick. (b-c) Our technique casts a ray from the touch point (laser or pen) perpendicular to the tablet surface to select all points along this ray (which length is defined by the Slicing-Volume thickness).

## 7.2 Related work

### 7.2.1 Occlusion in 2D/3D Selection

Object selection has been identified as a fundamental task in 3D user interfaces ([Mine, 1995](#); [LaViola Jr et al., 2017](#)) and one of the primary task in VR interaction ([Bowman et al., 1999](#)). Since 3D object selection is often affected by occlusion (e.g., density, target size, inter-object distance) ([Argelaguet et al., 2013](#)), multiple selection techniques have focused on solutions to overcome this limitation in selection tasks. For instance, Grossman et al. proposed a technique for selection in volumetric displays based on ray intersection (e.g., depth ray, lock ray, smart ray and flower ray) followed by a target disambiguation method consisting on moving a ray-aligned cursor controlled by dimensional device motion ([Grossman et al., 2006](#)). However, the selection in the disambiguation step still needs to be visually confirmed by the user, limiting their applicability to highly occluded environments, moreover, these techniques are focused to single object selection where the selection of a big number of targets will not be allowed. Olwal and Feiner introduced a flexible pointer that bends to avoid obstacles in a single-object selection task on a partially occluded view in a collaborative virtual environment ([Olwal et al., 2003](#)). This technique is highly dependent on the user's point of view limiting thus, the selection of totally occluded objects. Forsberg et al. replaced the ray with a cone-shaped volume to select small or distant objects, however, the cone aperture tends to over-select objects around the target ([Forsberg et al., 1996](#)).

Progressive refinement techniques have been explored for target disambiguation in locally dense spaces by iteratively splitting the search space into smaller hierarchical regions ([Kopper et al., 2011](#); [Bacim et al., 2013](#); [Mendes et al., 2017](#)). In these techniques, users perform repeated selections until the selection contains a single object. These techniques use several discrete steps to iteratively select an object within a group of interest, which can be tedious to reach the desired target, and it may be not suitable for highly occluded VEs ([Kopper et al., 2011](#)). Grossman et al. proposed a 2D area cursor technique called “bubble selection” which dynamically resizes a circular cursor to only contain one object at the time in a 2D selection task ([Grossman et al., 2005](#)). Further 3D versions were explored using a 3D volumetric cursor (sphere) instead ([Cockburn et al., 2004](#); [Vanacken et al., 2009](#); [Rosa et al., 2010](#)) adding transparency to the volumetric cursor to decrease occlusion over targets. However, these techniques are mostly focused on desktop applications and not suitable for VR. In mobile AR, DrillSample ([Mossel et](#)

al., 2013) is a two-step selection technique that first disambiguates occlusion by showing users an exploded view of all the objects near the select target. The user can then select the target in the second step. This technique, however, is only applicable to single-object selection.

Clustering methods have been also applied to address the density problem in selection tasks, such as the approach proposed by Lingyun Yu et al. where selection tools based on propagation techniques in point cloud data are used to select group points based on user draws (e.g. using 2D lasso tool) (Yu et al., 2012, 2016). Similarly, Shan et al. proposed a selection method in dense data sets using 2D circle and polygon tools to select regions of data where a clustering step is based on a voxelization of the selected region to increase clustering accuracy (Shan et al., 2014). These are powerful techniques that allow users to select 3D data from 2D inputs however, these techniques are in the frame of data visualization, as they are constrained to desktop or touchscreen monitor applications where a single mode gesture interaction for selection is allowed. In contrast, spatial interactions (e.g., hand gestures, controllers, hybrid-interfaces) are relatively less explored to interact with dense VR data.

In VR, numerous techniques have been introduced that take advantages of the expressive mid-air gestures supported in most VR systems (Argelaguet et al., 2013). “Large Scale Cut Plane” (Mossel et al., 2016) and “Yea Big, Yea High” (Jackson et al., 2018) allow users to use mid-air gestures to define a slicing plane in VR. Slicing plane is a well-known technique to reduce occlusion and filter data in the scientific visualization community (Obermaier et al., 2015). However, these techniques do not explore additional interaction modes and they were designed mainly for selecting single or partial objects (e.g., a vase or small patches of a 3D mesh), thus they may be not suitable for multi-target selection in dense environments. In contrast, our work explores the more challenging multi-target selection task in a dense VR environment. We explore a technique that maps the 3D selection task into a hybrid workflow that combines both expressive mid-air and grounded tablet interactions.

## 7.2.2 Selection using Mobile Devices in Mixed Reality

### 7.2.2.1 Visualization and Exploration

A tablet, when tracked in VR, can provide numerous powerful add-ons to how users interact with the VR environment (Bhaskar Surale et al., 2019). Most notably is the use of the tablet screen as an additional tool to support navigation and exploration of the



environment. In CAVE systems, Aspin et al. ([Aspin et al., 2007](#)) used a tracked tablet and pen to explore complex 3D models. Navigation is a prerequisite of selection in large-scale environment. Some techniques combine both navigation and selection on the tablet. Madeiros et al. ([Medeiros et al., 2013](#)) mapped finger gestures on tablet to enable selection and manipulation actions. Kim et al. ([Kim et al., 2008](#)) introduced a “finger walking in place” technique for navigation in VR using a tablet touch input. These techniques, however, only let users use 2D touch gestures to perform these 3D tasks and did not explore how to integrate them with more expressive mid-air interaction.

Mid-air interaction is arguably the primary mode of interaction in VR. When combined with a 6DOF tracked tablet, most work only explores simple cases such as menu navigation or single-object selection Bornik et al., combined 2D pen input on tablet, and mid-air interaction by means of a 3D controller in VR to enable medical data exploration and manipulation ([Bornik et al., 2006](#)). In Augmented Reality (AR), Wang and Lindeman employed 3D controllers for mid-air selection and multitouch tablet input for menu navigation ([Wang et al., 2014](#); [Wang et al., 2015](#)). Reitmayr et al. proposed a hybrid AR system for pen and a tablet interaction allowing 2D-3D selection in collaborative tasks ([Reitmayr et al., 2001](#)). Dias et al. explored hybrid systems and enabled 3D selection by gaze recognition ([Dias et al., 2018](#)). Bhaskar-Surale et al. explored the design space for using a multi-touch tablet in solid modelling in VR ([Bhaskar Surale et al., 2019](#)).

However, these techniques are limited to single/dual object selection and they do not explore the problem of density. Our approach extends this research by focusing on more challenging multi-target selection tasks and contributes a technique that takes advantage of a seamless hybrid mid-air/tablet workflow to address density issues in VR selection.

#### 7.2.2.2 *Haptic Feedback and Stability*

The advantages of mid-air interaction come mainly from the high DOF of the interaction. In VR, with a tracked controller or hand gestures, a user can easily move her hand in mid-air to describe complex shapes and curvature ([Schkolne et al., 2001](#)). However, mid-air interaction is known to be imprecise and can cause fatigue over extend use ([LaViola Jr et al., 2017](#)). Many research has investigated complementing mid-air gestures with physical props to increase stability ([Poupyrev et al., 1998](#); [Schmalstieg et al., 1999](#); [Bowman et al., 2001](#); [Bowman et al., 2003](#); [Lucas, 2005](#)). Most work, however, focuses only on content creation tasks such as VR sketching ([Arora et al., 2017](#); [Arora et al., 2018](#)) and modelling ([Billinghurst et al., 1997](#); [Szalavári et al., 1997](#)). Ours explores the accuracy

issue of mid-air interaction in a new context (i.e., multi-target VR selection). We explore a new strategy to increase precision by reducing the complexity of the selection task into a hybrid selection workflow.

In summary, most of the previously mentioned selection techniques that employ hybrid metaphors (i.e., combine both mid-air and tablet) do not support fluent multimodal interaction. That is, they constrain the selection to only one modality at a time, either on mid-air or on the physical tablet surface (Benko et al., 2007; Strothoff et al., 2011; Medeiros et al., 2013; Giesler et al., 2014; Afonso et al., 2017; Besançon et al., 2017; Sundén et al., 2017). On the other hand, several mid-air techniques in VR/AR only focus on simple use cases such as menu navigation or single-object selection (Bowman et al., 2001; Wang et al., 2014; Wang et al., 2015; Arora et al., 2018; Bhaskar Surale et al., 2019). Our work contributes a selection technique that lets users seamlessly use both modalities to perform the selection task in a challenging dense VR environment.

### 7.3 Implementation and Setup

We implemented our system in Unity (version 2018.4.3), using the point cloud importer “Pcx” (Takahashi, 2017). An Oculus Rift and two 3D touch controllers were used for VR interaction. We use an “iPad Air 1” as a physical proxy for our system.

The main goal of our system is to explore a hybrid 2D-3D *tablet-in-VR* approach for improving precision in multi-target selection tasks for dense VEs i.e., leveraging both haptic feedback and exploration offered by typical 2D interaction on a tablet (unlike arbitrary mid-air interaction). To achieve this, our system’s key components are: i) *Slicing-Volume* and ii) 2D-3D mapping (mid-air & tablet).

#### 7.3.1 Slicing Volume

Inspired by slicing planes commonly used in MRI analysis to explore dense data sets, our *Slicing-Volume* (SV) defines a specific region in the 3D space to be mapped onto a 2D tablet view in VR, for multi-object selection tasks (Yu et al., 2012, 2016) (see **Figure 7.1**). Contrary to slicing planes that usually visualize the cross-section of the data (the slice), *Slicing-Volume* uses the orthographic projection view of a virtual camera (adjusting projection size and clipping planes’ positions) to capture the content enclosed within its volume, but not the neighbouring elements. The rendered view on a virtual tablet surface within the VE, allows users to have a visualization tool to filter out elements that surround the desired target selection.

The *Slicing-Volume* provides a quick and flexible way to specify the initial selection, which can be refined i.e., it can be grabbed, translated and rotated by pressing the grip button on the right controller, to be easily placed and oriented by users during the selection task. Its dimensions can be dynamically adjusted using the joystick on the left controller (see **Figure 7.2**) allowing users to customize the 3D space mapped on the current virtual tablet view e.g., increasing the thickness of volume facilitates the selection of a larger set of points, while decreasing the thickness facilitates precise selection of occluded points (**Figure 7.2**). This *Slicing-Volume* then enables an exploration tool for highly dense data sets, that we employ for more precise selection.

During a selection task, the volume is initially attached to a virtual tablet in the VE, which in turn is attached to the left controller though a 3D-printed holder (see **Figure 7.3**), for free translation/rotation using the hand position and orientation. Then, to fix the volume in a specific position in the VE, the user presses the trigger button on the left controller to decouple the volume from the tablet.

#### 7.3.1.1 Selection Modalities

We designed two different modes to use our *Slicing-Volume*: (a) combined with a *Virtual Tablet* only (i.e., providing an exploration tool but not haptic feedback) and b) combined with both a *Virtual* and a *Real* tablet (i.e., providing both exploration and haptic feedback).

*Real Tablet* mode: We tracked a real tablet (iPad air) that was aligned to the virtual one allowing touch interaction and a physical limit using its touchscreen in VR. Selection in physical tablet mode enables a *pen-and-tablet* metaphor, a 10cm touch pen was attached to right controller allowing bimanual interaction (see **Figure 7.3**).



**Figure 7.3** Pen-and-tablet metaphor. Real Tablet attached to the left controller through an adjustable 3D-printed holder (2 joints). Physical stylus attached to the right controller for touch input on the tablet. The Slicing volume is attached to the tablet in the VE.



*Virtual Tablet* mode: Since this mode does not involve a physical tablet for selection, we employ a virtual laser attached to right controller, using a ray-casting technique to interact with the *Virtual Tablet* surface allowing also a *pen-and-tablet* metaphor, but without haptic feedback (**Figure 7.8**).

It is worth mentioning that the physical tablet view is never used for actual visualization (the user is wearing a Head-Mounted Display and immersed in a VE), but only for haptic feedback and touch input. Then, the user always explores the 3D space using the *Virtual Tablet* view, that can be accompanied by a physical tablet (aligned to the virtual one) or not, depending on the selection modality (see **Figure 7.8** for a comparison of the two selection modalities).

For selecting elements on the tablet view, the user brushes with the pen/laser on the tablet surface (either virtual or physical) and the strokes are mapped to the *Slicing-Volume*, i.e., the elements in the tablet view colliding with either the laser pen or the real pen are selected and projected along the *Slicing-Volume* thickness (see **Figure 7.2(b-c)**). Finally, the selection done on the tablet is updated in the 3D space.

These two modes (virtual and physical) allowed us to independently compare the two primary benefits of *tablet-in-VR* approaches suggested in the literature (i.e., extra visualization and stability), and thus explore the elements that contribute to improve accuracy of selection (if any). Selection events in both modes are triggered by holding down the trigger button on the right controller while brushing on the tablets' surface. Deselection events are also considered and triggered by pressing the “B” button on the right controller.

### 7.3.2 3D-2D Mapping (Mid-air & Tablet)

To map 2D strokes from the tablet view to a 3D volumetric space, we first considered the initial size of the *Slicing-Volume* frontal face to be the same size and aspect ratio as the tablet screen (15.5cmx21cm), but extending the thickness by the larger tablet side, then our selected 3D space unit size  $U(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{R}_3$  is  $\{15.5\text{cm}, 21\text{cm}, 21\text{cm}\}$ , about  $6,868\text{cm}^3$ . The size  $U$  is the initial size of the *Slicing-Volume* in the virtual environment with scale = 1. As the virtual camera is attached to frontal face of the *Slicing-Volume*, this face represents the main plane to be rendered to the 2D tablet view.

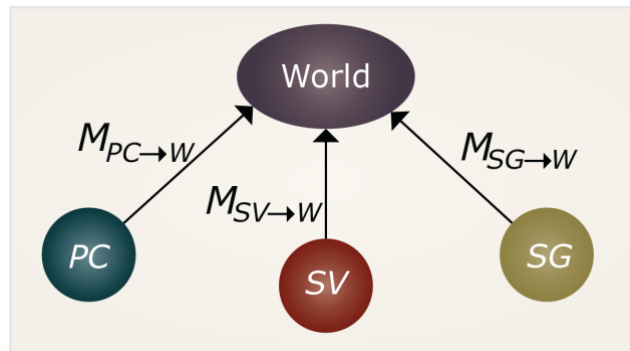
The system uses *compute shader* to process the information from the elements contained within the volume boundaries in each frame. All the elements intersecting the *Slicing-Volume* are highlighted in white (blended default point colour in the point cloud with a pure white), to be easily visualized from the tablet view, as well as in the VE.

The position of the elements in the volume are projected onto the main plane (clipping plane) which allows us to quickly detect the collisions from the tablet strokes (by pen/laser selection) and highlight the collided (selected) elements in green (blending the default point colour in the point cloud with a pure green) to indicate selection. These highlights (green and white) helps in the visual exploration of the current state of the points that belong and surround the target in the volume from either tablet and VE views.

### 7.3.3 Formal Description

In order to formally describe our system, we define the key components of our system: i) the point-cloud container **PC**, which contains the point cloud information (local points' position and colours) and serves as an anchor between the virtual world and the point cloud data, ii) the slicing volume **SV**, i.e., a 3D volume in the virtual world used to define a specific section in the 3D space to be mapped to a tablet surface allowing direct interaction in the virtual space (3D) from tablet touch inputs (2D), iii) the selection group (**SG**) which similarly to **PC** is a container in the virtual world that serves as an anchor in the world for the selected points only, enabling users to move and explore the selected elements independently from the rest of the points in **PC**. We will discuss these key elements in more detail next.

To formally describe the interaction with the point cloud, we use right-hand systems of reference, homogeneous coordinates (i.e., 3D points in A's coordinates as  $\mathbf{P}_A(x, y, z, 1) \in \mathbb{R}^4$ ) and homogeneous transformation matrices ( $\mathbf{M}_{A \rightarrow B} \in \mathbb{R}^{4 \times 4}$  to convert coordinates from A to B).



**Figure 7.4 System node graph.**

If we analyse the system based on the node graph shown in **Figure 7.4**, we identify 4 main nodes: Point Cloud (**PC**), Slicing Volume (**SV**), Selection Group (**SG**) and the virtual World node (**W**). We can explain then the system behaviour in function of transitions of points between nodes (each node has its own system of reference). When selecting a point  $p_i$  from the point cloud (**PC**), the point  $p_i$  is moved to the selection group node (**SG**) and the current position in **SG** system of reference is computed by:

$$p_{sg\ k} = (M_{SG \rightarrow W})^{-1} \cdot M_{PC \rightarrow W} \cdot p_i \quad (7.1)$$

Where  $k$  is the point index in the **SG** node,  $p_i$  is the recently selected point from **PC**, while  $M_{PC \rightarrow W}$  and  $(M_{SG \rightarrow W})^{-1}$  are the transformation matrices to move the point through the nodes, from **PC** to **W** and from **W** to **SG**.

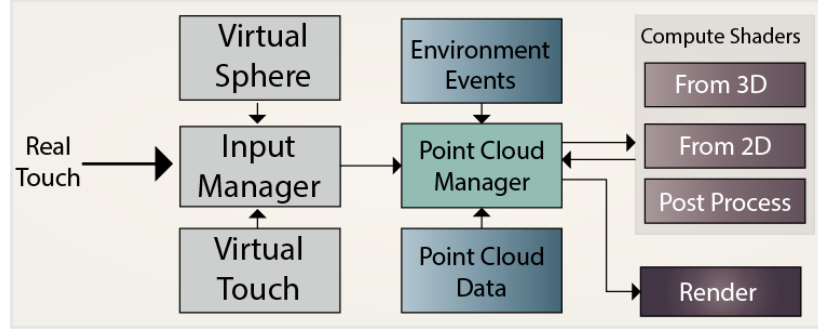
#### 7.3.3.1 Slicing-Volume and Point Cloud Interaction

We can describe the interaction between the **SV**, the points in **PC** and the selected points in **SG** more formally if we let  $\mathbf{P}=\{p_1, p_2, \dots, p_n\}$ ,  $\mathbf{V}=\{v_1, v_2, \dots, v_n\}$ ,  $\mathbf{G} = \{s_1, s_2, \dots, s_n\}$  be the set of points in the point cloud (**PC**), the set of positions in the slicing volume (**SV**) and the set of points in the selection group (**SG**), respectively. Then, we can define a sub-set  $\mathbf{Iv}$  that represents the intersection between  $\mathbf{P}$  and  $\mathbf{V}$  as  $\mathbf{Iv}=\mathbf{P} \cap \mathbf{V}$ , where  $\mathbf{Iv} \subseteq \mathbf{P}$  and  $\mathbf{Iv} \subseteq \mathbf{V}$  are the points in the point cloud inside the slicing volume (**SV**).

The **SV** size can be adjusted by two factors, i) aspect size and ii) thickness. Aspect size refers to the x,y plane (tablet frontal face), and thickness refers to the z-axis (adjustment of depth). The change ratio ( $\Delta R$ ) was identified empirically through a small pilot study with 6 participants where the task was to play with manually tuned  $\Delta R$  to find the one or the ones they felt more suitable for the task. The pilot results show 2 sets of thresholds and  $\Delta R$ s (one for each aspect ratio and thickness) that we use in our system.

#### 7.3.3.2 Real-Time GPU-based Solution

Processing big amounts of data in the CPU drastically drops the FPS down. We then used *ComputeShaders* and standard shaders to share/pass the interaction with the point cloud to the GPU (e.g., collisions between Unity game objects and individual points in **PC**). This allowed us to interact in real-time with big data sets (~1 million points). In our system, we implemented three main *ComputeShaders* (*selectionIn2D*, *selectionIn3D* and *SVIntersection*) to manage the point cloud interaction, but only one is used per Unity update call. **Figure 7.5** shows a diagram that represents the behaviour of our system from the GPU perspective.



**Figure 7.5** The general behaviour of the system.

### 7.3.3.3 Selection in 2D-ComputeShader

Selection in 2D is computed by using the position of the touch input (from the real/virtual tablet) in **SV** coordinates, and the points' positions and slicing volume position in **W** coordinates. To compute the selection from 2D, we considered  $\mathbf{T}=\{t_1, t_2, \dots, t_n\}$  to be the set of positions in the propagated touch in **SV** coordinates (see **Figure 7.2**). Then, 2D *ComputeShader* is used to compute: i) the set of elements intersecting the slicing volume (**SV**) and ii) the set of elements intersecting the propagated touch (**T**). If we assume that  $\mathbf{Iv}=\mathbf{P} \cap \mathbf{V}$  is the intersection between the elements in **PC** and **SV**; and  $\mathbf{It} = \mathbf{P} \cap \mathbf{T}$  is the intersection between the **PC** and the propagated touch **T**, we can then compute:

i) The new point's position in the set **V** (the intersecting points in **SV** coordinates) as:

$$\mathbf{p}_{vj} = \cdot (\mathbf{M}_{SV \rightarrow W})^{-1} \cdot \mathbf{M}_{PC \rightarrow W} \cdot \mathbf{p}_i, \quad \forall \mathbf{p}_i \in \mathbf{Iv} \quad (7.2)$$

ii) The new point's position in the selection group (set of points selected) coordinates as:

$$\mathbf{p}_{gj} = (\mathbf{M}_{SG \rightarrow W})^{-1} \cdot \mathbf{M}_{PC \rightarrow W} \cdot \mathbf{p}_i, \quad \forall \mathbf{p}_i \in \mathbf{It} \quad (7.3)$$

#### **Pseudocode 1: 2D volume intersection and selection:**

```

For all points in the PC
    action = readAction;
    touch = inputTouch;
    p = pointInPCCoord;
    pInWorldCoord = PCtoWorldMat * p;
    pInVolumeCoord = (VolumeToWorld)-1 * pInWorldCoord;
    pIsInVolume = IsInVolume(pInVolumeCoord);
    if(pIsInVolume)
        highlight(p);
    pIsSelected = isColliding(pInVolumeCoord, touch);
    if(pIsSelected)
        doAction(p, action);

```

As a result of **Eq. 7.2** and **7.3**, all the points in the set **P** intersecting with set **V** will be moved to the node **SV** (slicing volume node) and all the points intersecting with the

propagated touch (set **T**) will be moved to the node **SG**. The general behaviour of the *selectionIn2D* shader is also explained in the pseudocode 1.

#### 7.3.3.4 Selection in 3D-ComputeShader

Unlike selection in 2D (where selection events are activated by actual touch on the real tablet or collision from laser pointer on the virtual tablet), selection in 3D uses a position of a floating 3D sphere (used as pointer to select/deselect on real/virtual tablet) in world coordinates, and the points' positions in **W** coordinates. As only one *ComputeShader* is used per Unity update call, the intersection **Iv** is also computed on this shader. To compute elements' selection, we consider **S**={ $s_1, s_2, \dots, s_n$ } to be the set of position in the selection sphere in **W** coordinates.

The 3D *ComputeShader* computes: i) the set of elements intersecting the slicing volume (**SV**) and ii) the set of elements intersecting the selection sphere (**S**). Taking the intersection of the elements **Iv**=**P**∩**V** (between the **PC** and the **SV**) and **Is** = **P**∩**S** (between the point cloud and the selection sphere), we then compute:

- i) The new point's position in set **V** using the **Eq. 7.2**.
- ii) The new point's position in the selection group (set of points selected) coordinates for all element in the intersection sub-set **Is** as:

$$p_{gj} = (M_{SG \rightarrow W})^{-1} \cdot M_{PC \rightarrow W} \cdot p_i, \quad \forall p_i \in Is \quad (7.4)$$

As a result of **Eq. 4**, all the points intersecting with **S** will be moved to the node **SG**. The general behaviour of the *selectionIn3D* shader is also explained in the *pseudo-code 2*.

#### **Pseudocode 2: 2D volume intersection and 3D selection:**

```

For all point in the PC
    action = readAction; //select / deselect
    spherePos = SpherePosInWorld;
    p = pointInPCCoord;
    pInWorldCoord = PCtoWorldMat * p;
    pInVolumeCoord = (VolumetoWorldCoord)-1*pInWorldCoord;
    pIsInVolume = IsInVolume(p);
    if(pIsInVolume)
        highlight(p);
    pIsSelected = isColliding(pInWorldCoord, spherePos);
    if(pIsSelected)
        doAction(p, action);

```

Summirizing, in this implementation, all the *ComputeShaders* described are called by Unity and executed on the GPU along with standard shaders in order to enable object-user interaction with large amounts of elements (~1 million objects) in real-time. However, only one *ComputeShader* is called per update call, to ensure correct assignation of elements' states (e.g., selected) based on the current user's action.

## 7.4 User Study

We conducted a within-subjects user study in VR to test our system. Our study aimed at (1) explore whether our approach improves user performance in selection tasks and (2) gain insight about the cause of the potential benefits obtained (i.e., visualization vs haptic feedback). We asked participants to select a specific set of target points from a dense point cloud environment in VR (see **Figure 7.6(a)**).

As we are interested in understanding the benefits of having a tangible tablet for this task in VR, we compared our two selection modalities (see **Figure 7.8**): “pen selection” using both *Virtual* and *Real* tablets (i.e., involving *Slicing-Volume* exploration tool and haptic feedback) and “laser selection” using the *Virtual Tablet* only (i.e., involving *Slicing-Volume* exploration tool but not haptic feedback). A baseline condition was also included in the study, consisting of a standard 6-DOF brushing-based selection technique i.e., *Mid-air* only selection using one 3D controller without the use of *Slicing-Volume* tool nor tablet modalities. The detailed design of these conditions is described in the Experimental Conditions section and depicted in **Figure 7.7**.

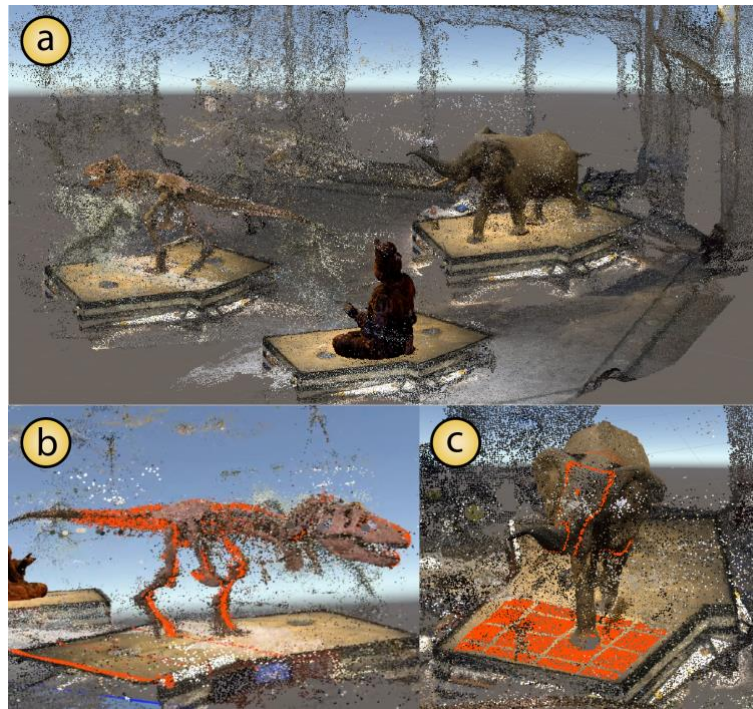
### 7.4.1 Participants

Twelve participants (2 females, mean age=28.08 years old, SD=4.62) took part in our study. They were all right-handed with no previous experience on VR. The recruitment criteria only included having normal or corrected-to-normal vision and having no injuries on their hands and/or arms. The local ethics committee approved this study. A full session took about 60min and participants were not paid for their participation.

### 7.4.2 Task Point Cloud Model

We designed a dense point cloud environment for the study. Our point cloud model consists of 984,613 points (~1 million). The model shows a museum with three sculptures (i.e., a statue, a T-Rex and an Elephant) as shown in **Figure 7.6**. The statue was used only for a training stage. Then, the T-Rex and Elephant sculptures were used for the main tasks. The whole point cloud (containing the three sculptures) was rendered in a 1.5m x 1.5m x 1m volume, each model's dimensions were then about 80m x 45m x 50m.

On these sculptures, we highlighted in red a set of target points strategically distributed to have two different levels of difficulty in the selection task i.e., this difficulty depends on the level of occlusion in each model (see **Figure 7.6(b-c)**). We defined the occlusion level by the point density per space unit (**U**). The Elephant model has an average density of 22,597 points/**U** (level of occlusion), containing 7,293 target points (see **Figure 7.6(c)**). Then, the T-Rex model has an average density of 42,687 points/**U**, containing 19,392 target points (see **Figure 7.6(b)**). This means that the density of the T-Rex model was almost two times more occluded and therefore more challenging than the elephant model.



**Figure 7.6 Virtual Environment used in the user study(a) (~1million points). T-Rex model highlighting the distribution of target points (19,392) (b). Elephant model and its distribution of target points (7,293) (c).**

The rationale behind our scenario and manual task design lays on two main factors. First, the manual selection of target points allowed us to make clear the difference in selection difficulty in terms of occlusion between tasks. That is, the target points in the T-Rex model is in a much denser region and is much more difficult than the Elephant model (as shown in **Figure 7.6(b-c)**). Second, we distributed the target points such that they do not cover specific parts of the model (e.g., one leg or the head). While this decision makes our tasks less realistic, it helps us ensure the tasks are equally challenging toward all of the conditions. In our pilot testing, we found that if the target points are specific (e.g., we placed the target points covering only the T-Rex's head), the Slicing-Volume technique can capture all of the targets in a short time with high accuracy. This does not fit our study

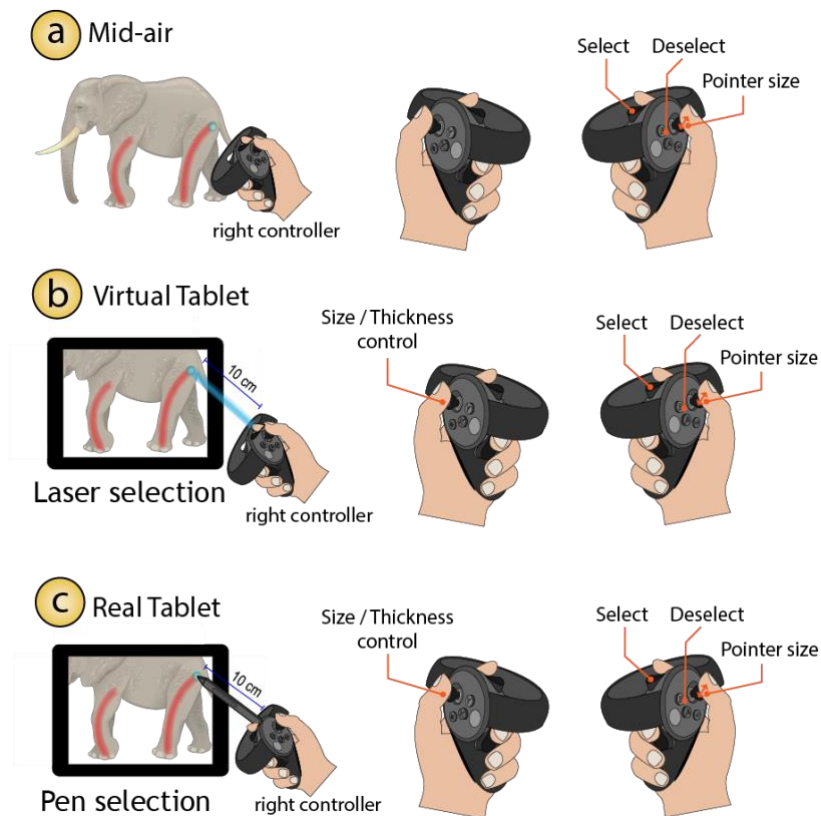


goal, i.e., the aim of our design was (1) make a fair comparison between all the conditions and (2) explore when our hybrid system could be more useful (i.e., for highly occluded or mildly occluded models), being level of task challenge clearly distinguishable.

### 7.4.3 Experimental Conditions

We designed our three experimental conditions to be as similar as possible so that measured effects are exclusively due to the interaction technique (*Mid-air*, *Virtual Tablet* and *Real Tablet*) instead of other parameters. In all of these three conditions, the main actions (select, deselect, and pointer size adjustment) were mapped to the right controller. In the two conditions involving the *Slicing-Volume*, the left controller is used to adjust the volume dimensions (see **Figure 7.7**). Each participant selected the target points in the two models (Elephant and T-Rex) using the three conditions.

*Condition 1 (Mid-air only)*: This is our baseline condition. Participants were given only the right controller and asked to select the target points by brushing in mid-air. A floating sphere in front of the controller indicated the pointer to select/deselect. Points colliding with this sphere are then selected/deselected. The size of the pointer could be increased/decreased (with a minimum diameter of 0.7cm) to support varying levels of precision. Neither *Slicing-Volume* tool nor tablets were used in this condition.

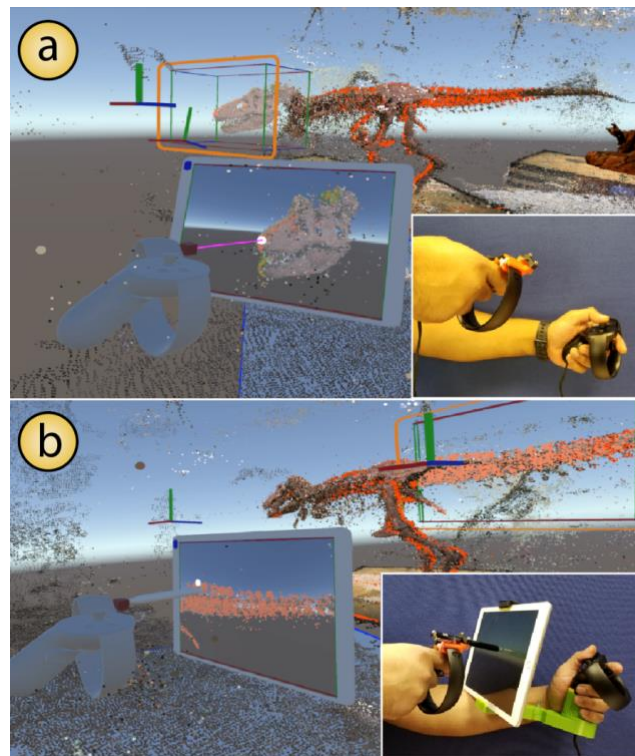


**Figure 7.7 Controller actions for the three interaction techniques.**



*Condition 2 (Virtual Tablet):* Participants were given two controllers and, in addition to the mid-air selection, they were also allowed to use the *Slicing-Volume* tool. They could place the volume (which was initially attached to the *Virtual Tablet* representation) in the desired area (see **Figure 7.1(a)**) and adjust its size and thickness to explore the models (see **Figure 7.1(b)**). The points enclosed within the volume were mapped to the *Virtual Tablet* view where participants could select the target points by “laser selection” modality (see **Figure 7.8(a)**). Selection/deselection events were enabled by the “laser” tip (0.7cm diameter) colliding with the *Virtual Tablet* surface. The selection in the *Virtual Tablet* was then propagated along the *Slicing-Volume* thickness and updated in the 3D model.

*Condition 3 (Real Tablet):* In addition to mid-air selection and *Virtual Tablet* representation with *Slicing-Volume* tool, participants were given an iPad air tablet which was attached to the left controller through an adjustable 3D-printed holder (see **Figure 7.8(b)**). A physical pen was then attached to the right controller (replacing “laser selection”). Selection/deselection events were enabled by the pen tip (0.7cm diameter) touching the *Real Tablet* touchscreen (see **Figure 7.8(b)**). The selection in the *Real Tablet* was then propagated along the *Slicing-Volume* thickness and updated in the 3D model.



**Figure 7.8 Selection modalities: (a) Laser selection: Participants select on the tablet view using a visible “laser” beam (10cm long) with no tangible feedback. (b) Pen selection: Participants select using an actual pen (10cm long) on the tangible tablet surface. The pen is also virtually represented.**

#### 7.4.4 Experimental Setup

Participants were asked to sit next to a semicircle-shaped table and adjust the chair height for comfort. They then wore an Oculus Rift headset (see **Figure 7.9**). Participants could rest the *Real Tablet* on the table to avoid fatigue. Participants could also translate/rotate the whole environment to interact with it while sitting (avoiding overstretching).

Subsequently, participants performed a training stage that lasted 15 minutes where they were allowed to get familiar with the system and practice the selection task (selecting target points from the statue model only). After the training, we presented the two models (T-Rex and Elephant) with their target points highlighted in red. The order of the conditions (*Mid-air only*, *Virtual Tablet* and *Real Tablet*) as well as the order of the models were counterbalanced across participants. Each model in each condition was completed in a maximum time of 10 minutes.

#### 7.4.5 Variables measured

We measured the accuracy of the selection based on the F1 score used by Yu et al. in a point cloud selection study (Yu et al., 2012, 2016). Briefly, the F1 score is computed from TP (true positive: number of target points correctly selected), FP (false positive: number of points incorrectly selected), and FN (false negative: number of target points that were not selected). Where F1 score is a weighted average of precision  $P = TP/(TP+FP)$  and recall  $R = TP/(TP+FN)$ . Then, F1 is computed as  $F1 = 2 \cdot (P \cdot R) / (P + R)$ . A F1 score of 1 indicates perfect selection performance and 0 otherwise. When we employ the term **accuracy** in our analysis, we refer to F1. Then, when we employ the terms **Over-Selection** and **Under-Selection** in our analysis, we refer to FP and FN respectively.



**Figure 7.9 Experimental setup.**

Time measures were also recorded, we measured **time of selection** (time that participants took in selection mode, while pressing trigger), **time of deselection** (time that participants took in deselection mode while pressing “B”) and **total time** which represents the Task Completion Time (TCT) in each model i.e., including the time participants took: (a) in selection mode, (b) in deselection mode, (c) observation (no mode activated) and (d) manipulating the volume. Additionally, we assessed participants’ user experience in each condition using a raw NASA TLX scale.

Since we are comparing experimental conditions (tablet modes) with controls (our baseline *Mid-air* only mode) we employed Standard Error of Mean (SEM) to represent error bars in our plots according to Rule 4 in (Cumming et al., 2007).

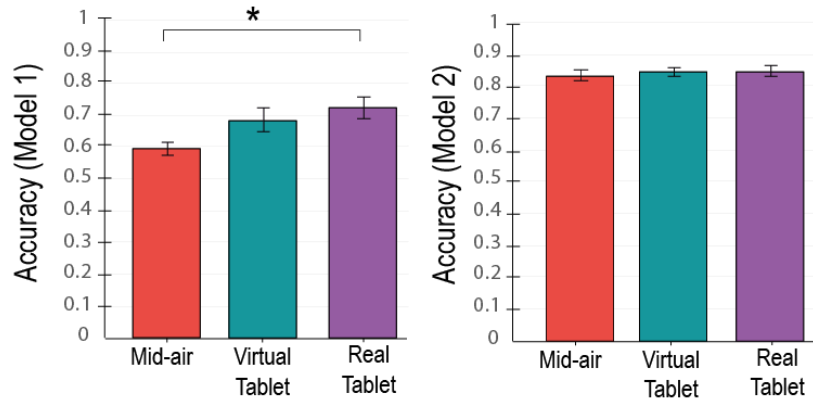
#### 7.4.6 Results

We conducted One-way Repeated Measures ANOVA tests in our analyses supported by Shapiro-Wilk Normality Tests in the three conditions as well as a Normal Q-Q plots and Mauchly’s test of sphericity (i.e., sphericity not rejected) to ensure that the normal distribution and homogeneity of the variance assumptions were not violated. Partial eta squared ( $\eta^2$ ) is reported as a measure of effect size, according to Cohen (Cohen, 1988), we refer to a value of 0.01 as a small effect, 0.06 a medium effect, and 0.14 or greater as a large effect size.

##### 7.4.6.1 Accuracy

A One-way Repeated Measures ANOVA was conducted to compare the effect of the interaction technique (i.e., *Mid-air only*, *Virtual Tablet* and *Real Tablet*) on participants’ accuracy of selection in Model 1 (T-Rex). Results showed significant effect of the interaction technique ( $F_{(2,22)}=3.27, p=0.017, \eta^2=0.557$ ). Comparison tests with Bonferroni correction showed a significant difference between *Mid-air* and *Real Tablet* conditions ( $p=0.039$ ) but we found no difference between *Mid-air* and *Virtual Tablet* conditions ( $p=0.135$ ). See **Figure 7.10**(left), for mean F1 scores.

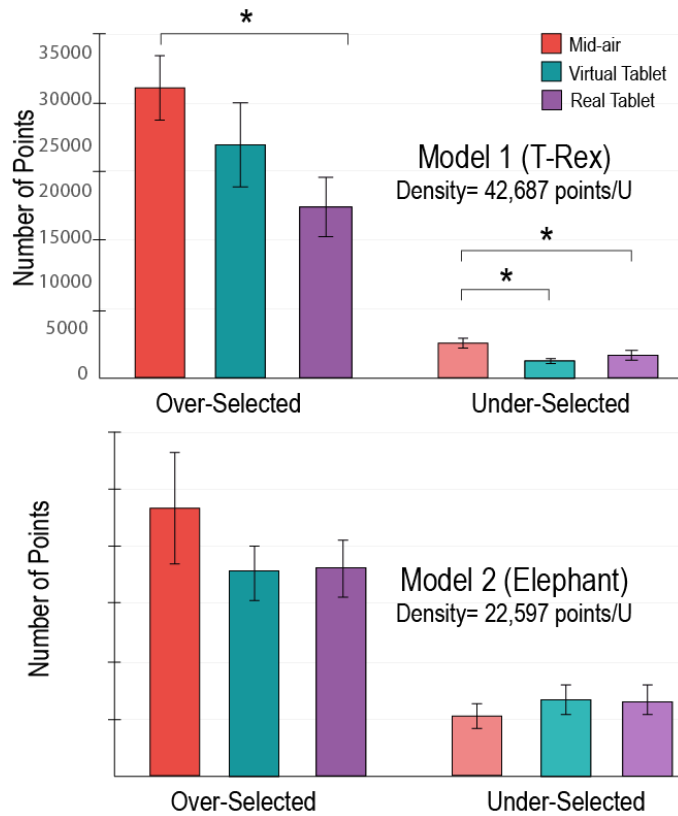
The data collected from Model 2 (Elephant) did not meet the homogeneity of the variance assumption ( $p=0.002$ ), and we therefore carried out a non-parametric One-Way Repeated Measures ANOVA (Related-Samples Friedman's Two-Way Analysis of Variance by Ranks). Results showed no significant effect of the interaction technique on participants accuracy of selection ( $p=0.92$ ). See **Figure 7.10**(right), for mean F1 scores.



**Figure 7.10 Results on Accuracy (F1) for Model 1 (left) and Model 2 (right). Error bars represent SEM. \*= $p < 0.05$ .**

#### 7.4.6.2 Over-Selection and Under-Selection

A One-way Repeated Measures ANOVA was then conducted to compare the effect of the three interaction techniques on participants' Over-Selection and Under-Selection in Model 1 (T-Rex). Results showed significant effect of the interaction technique on Over-Selection ( $F_{(2,22)}=4.04$ ,  $p=0.05$ ,  $\eta^2=0.447$ ). Comparison tests with Bonferroni correction showed a significant difference only between *Mid-air Only* and *Real Tablet* conditions ( $p=0.04$ ).



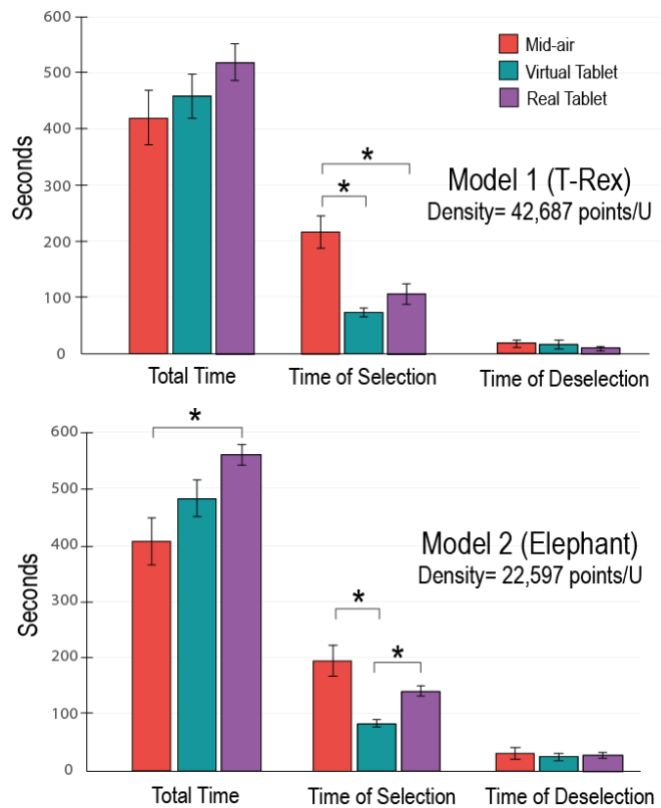
**Figure 7.11 Results on Over-Selection (FP: false positive) and Under-Selection (FN: false negative) for Model 1 (top) and Model 2 (bottom). Error bars represent SEM. \*= $p < 0.05$**

Results also showed a significant effect of the interaction technique on participants' Under-Selection in Model 1 ( $F_{(2,20)}=4.04$ ,  $p<0.001$ ,  $\eta^2=0.817$ ). Comparison tests with Bonferroni correction showed a significant difference between both *Mid-air only* and *Virtual Tablet* conditions ( $p<0.001$ ) and between *Mid-air only* and *Real Tablet* conditions ( $p=0.016$ ) (see **Figure 7.11**(top)).

The same test was conducted for Model 2 (Elephant). However, no significant effect was found of the interaction technique on Over-Selection ( $F_{(2,22)}=0.546$ ,  $p=0.596$ ,  $\eta^2=0.098$ ) nor Under-Selection ( $F_{(2,22)}=0.460$ ,  $p=0.644$ ,  $\eta^2=0.084$ ), (see **Figure 7.11**(bottom)).

#### 7.4.6.3 Time

A One-way Repeated Measures ANOVA was again conducted, to compare now the effect of the interaction technique on participants' time of selection, time of deselection and total time in Model 1 (T-Rex). Results showed significant effect only for time of selection ( $F_{(2,22)}=18.47$ ,  $p<0.001$ ,  $\eta^2=0.787$ ). Comparison tests with Bonferroni correction showed a significant difference between both *Mid-air only* and *Virtual Tablet* conditions ( $p=0.001$ ) and between *Mid-air only* and *Real Tablet* conditions ( $p=0.01$ ), (see **Figure 7.12** (top)).



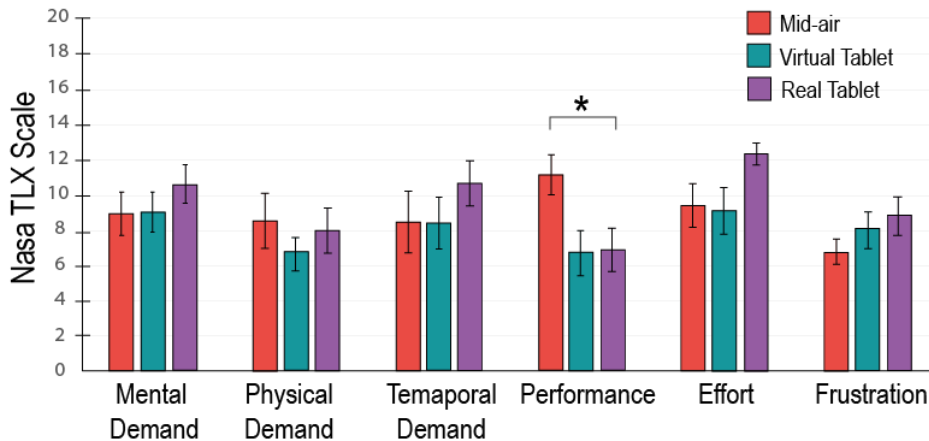
**Figure 7.12 Results on Time in Model 1 (top) and Model 2 (Bottom). Error bars represent SEM. \*= $p<0.05$ .**

The same test was conducted for Model 2 (Elephant). Significant effect of the interaction technique on total time was found ( $F_{(2,22)}=12.29$ ,  $p<0.002$ ,  $\eta^2=0.711$ ). Comparison tests with Bonferroni correction showed a significant difference only between *Mid-air only* and *Real Tablet* conditions ( $p=0.002$ ). Significant effect was also observed for time of selection ( $F_{(2,22)}=12.29$ ,  $p<0.002$ ,  $\eta^2=0.711$ ). Comparison tests with Bonferroni correction showed a significant difference between both *Mid-air only* and *Virtual Tablet* conditions ( $p=0.003$ ) and between *Virtual Tablet* and *Real Tablet* conditions ( $p<0.001$ ).

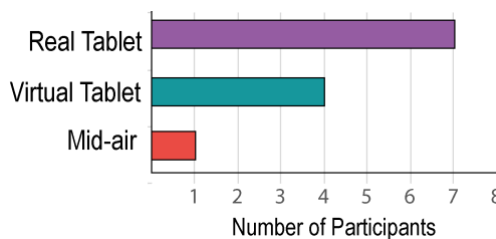
However, no significant effect of the interaction technique on deselection time was observed for Model 2 ( $F_{(2,22)}=0.15$ ,  $p<0.85$ ,  $\eta^2=0.030$ ), (see **Figure 7.12** (bottom)).

#### 7.4.6.4 User Experience

We also analysed participants' subjective experience using a raw NASA TLX scale. A One-way Repeated Measures ANOVA was conducted to compare the effect of the interaction technique on participants' user experience, (specifically on: *mental demand*, *physical demand*, *temporal demand*, *performance*, *effort* and *frustration*) in both Models. Results showed significant effect of the interaction technique only on *performance* ( $F_{(2,22)}=3.99$ ,  $p=0.05$ ,  $\eta^2=0.444$ ). Comparison test with Bonferroni corrections showed a significant difference only between *Mid-air Only* and *Real Tablet* conditions ( $p=0.04$ ). See **Figure 7.13** for mean scores.



**Figure 7.13 Results on user experience in both models. Error bars represent SD.  $\ast=p<0.05$ .**



**Figure 7.14 Participants' preferred interaction technique.**



## 7.5 Discussion

The results from our user evaluation show that our hybrid interaction technique actually improved participants' accuracy of selection in a highly occluded task environment. However, we found that the participants' accuracy depends on the occlusion level of the task. Recall that our two models differ in terms of occlusion: Model 1 T-Rex (42,687 points/U) is more occluded than Model 2 elephant (22,597 points/U). In Model 2, the difference in participants' accuracy is minimal, suggesting that the hybrid workflow may not be as advantageous for improving the selection performance in this scenario (see **Figure 7.10(right)**).

However, in Model 1, we found that participants in the *Real Tablet* condition were significantly more accurate than in the *Mid-air* condition. This finding suggests that the reduction of multi-target selection into a hybrid workflow (that combines mid-air gestures, 2D selection, and physical proxies) through using a *Slicing-Volume* was more beneficial than the baseline *Mid-air* only condition during the selection task. Although the *Virtual Tablet* condition also outperforms the *Mid-air* condition, the difference was not found to be statistically significant. This result suggests that having only the additional visualization tool (*Virtual Tablet* view) may not be enough to significantly improve accuracy compared with *Mid-air* condition in a highly occluded environment. This suggests that the physical surface of the *Real Tablet* is clearly beneficial to the task due to the extra haptic and stability support. Furthermore, these results still hold when considering Over-Selection and Under- Selection measures of Model 1.

Under-Selection refers to target points participants missed during the task. When a scene is highly occluded, it affects a user's ability to locate the target ([Argelaguet et al., 2013](#)). Thus, participants trying to select in an occluded environment may suffer more from Under-Selection. In our study, during both *Virtual* and *Real Tablet* conditions, participants significantly under-selected less compared with the *Mid-air* only condition. This suggests that participants benefited from the hybrid workflow, which used the *Slicing-Volume* visualization to help disambiguate occluded targets. However, our results showed no benefit of the tablet conditions (*Virtual* and *Real*) when selecting Model 2. In summary, in terms of Under-Selection, having an extra visualization (given by the *Slicing-Volume* tool) helped to improve participants' performance, rather than having haptic feedback (given by the physical tablet surface) for the most occluded model only.

Over-Selection refers to points participants unintentionally selected. This measure reflects the participants' ability to trace the target points to make the selection. As shown in our results, participants in the *Mid-air* only condition made much more Over-Selection than in the other two conditions, supporting that mid-air gestures are known to be inaccurate (Arora et al., 2017). However, we found that only during the *Real Tablet* condition participants over-selected significantly less than during the *Mid-air* only condition. This suggests that participants benefited from the physical tablet surface thanks to its stability and the selected more accurately. However, again our results showed no benefit of the tablet conditions (*Virtual* and *Real*) when selecting Model 2. In summary, in terms of Over-Selection (in a highly occluded environment), having haptic feedback (given by the physical tablet surface) helped to improve participants' performance, rather than having an extra visualization (given by the *Slicing-Volume* tool) for the most occluded model only.

These findings further show that our technique complements well the inaccuracy of mid-air gestures and can act as an effective tool to inspect and select target points in highly occluded environments.

It is worth mentioning that, the *Real Tablet* comes with extra weight, which may have affected participants' ability to quickly execute actions in the study. For instance, in the measured time (**Figure 7.12**), we found that participants spent more total time in the *Real Tablet* condition. This could be due to the extra manoeuvres they had to execute to look at the tablet and adjust the volume thickness. However, total time was only significantly higher when selecting Model 2 (involving lower points density). We interpret these results as our hybrid system is more effective for higher dense models (e.g., our Model 1) i.e., improving accuracy without significantly increasing the TCT.

One interesting observation in our results is that overall participants took more time of selection (i.e., in "selection mode" while pressing trigger) in the *Mid-air* only condition while selecting both models (**Figure 7.12**). That is, even when participants took more total time (i.e., TCT) during the tablet conditions, they spent significantly more time in "selection mode" during the *Mid-air* only condition. We interpret these results as participants could select target points faster and felt more confident about their selection (i.e., spending less time selecting) due to both stability and extra visualization provided by the *Virtual* and *Real* Tablets.



Regarding participants' subjective UX collected using a raw NASA TLX scale, we found no effect of the interaction techniques on *mental demand*, *physical demand*, *temporal demand*, *effort and frustration*. Although we observed trend showing higher scores for the *Real Tablet* condition, it was found non-significant. However, we found that participants perceived significantly better performance during the task when using the *Real Tablet* (see **Figure 7.13**).

There are certain trade-offs in the design of the *Real Tablet* that may affect the user experience. On one hand, the physical sensation of using the *Real Tablet* helped participants feel more grounded during the task. When asked about the preferred interaction technique, the majority of the participants preferred the *Real Tablet* (see **Figure 7.14**). For example, P5 commented “the feeling of having a physical limit to select makes it more real”, P7 mentioned “a physical surface in selection is useful” and “the *Virtual Tablet* feels less natural than the *Real Tablet*”. P8 said “the *Virtual Tablet* allows me to select faster, but I may tend to over-select” and “the *Real Tablet* gives the tactile cues to link with the real world, that feels cool”.

Overall, our hybrid approach showed positive results suggesting relevant benefits to be used in highly occluded VR models (e.g., editing: point-cloud models or dense 3D painting). The reduction of the multi-target selection into the tablet space through the *Slicing-Volume* tool enable users to manage occlusion and achieved better selection performance. It also complements the imprecise nature of mid-air selection, allowing users to select with higher selection accuracy. While hybrid approaches have been previously explored in the literature and suggested to provide advantages in VR tasks, in this chapter we provide an analysis of the benefits of this novel hybrid workflow in a dense VR selection task with different levels of density and occlusion. With our results, we expect to contribute insights to consider when designing *tablet-in-VR* approaches, particularly those involving dense environments.

## 7.6 Limitations and future work

For experimental purposes we kept all conditions simple with only few tools enabled. However, for future work we will add more tools to our system such as lasso, undo, delete, and save. A limitation of our study is that it was done with only a small set of novice VR users. Given the positive results from this preliminary analysis, a direction for future work is to evaluate our system with professional 3D designers to gain deeper insight of the

benefits of our hybrid system in more application scenarios such as actual editing e.g., point cloud, dense painting and sculpting tasks.

## 7.7 Conclusion

We presented a novel hybrid interaction technique called *Slicing-Volume* for selection in highly occluded point cloud models in VR. Our approach provides users with a 3D volume visualization that can enclose target objects in mid-air. Users can use this volume to inspect and filter out potential target points for selection. These points are then mapped to a 2D tactile tablet surface so that users can select with haptic feedback given by the tablet surface. We tested our system in a user study and found that our interaction technique improved the accuracy of selection in a highly occluded point cloud model in VR. Our results also provide insights into the specific benefits of our technique. We found that the mapping of the selection region into a 2D tactile surface on the tablet is generally the most beneficial when it comes to aiding the selection. The additional volume visualization is also favourable to help users filter occluded points, but only when used together with the tactile tablet surface. Overall, these findings show that our technique can act as an effective tool to help users inspect and select target points in dense VR environment.

## Chapter 8

# Conclusions

The main goal of this thesis was to explore *human factors* to help in the generation of more natural 3DUIs that better benefit from human capabilities, but also considering human limitations. To achieve this goal, we combined *human factors*, mathematical formalizations and computational methods (taking advantage of current computational power). However, due to the very broad extension of the 3DUI field, it was not possible to provide an extensive exploration over the full scope in this thesis. Therefore, to explore the potential of our approach, we based our analysis on the stages of interaction proposed by (Molina-Masso et al., 2008) shown in **Figure 2.4** i.e., the hardware-ITes mapping, ITes and ITas.

Designing 3DUIs involves many challenges that may represent important limitations within this interaction stages in VR, which can be human-related (e.g., human capabilities) (Stanney et al., 1998) or hardware/software-related (e.g., tracking devices and techniques). Throughout the exploration in this work, we addressed some of the current limitations involved in VR produced by the complexity of 3D interaction. Some of the main issues identified in the literature regarding the interaction stages were: (1) the complexity of 3D *input* mappings in gestural interaction, (2) fatigue in object *manipulation* produced by mid-air interaction, (3) space limitations in VR *navigation*, (4) *Drift* effects produced by *scale-adaptive* VR navigation and (5) low accuracy in multitarget 3D object *selection* due to occlusion and lack of stability. Next, we describe our contributions by addressing these limitations in each chapter.

## 8.1 Individual Chapter Contributions

Along five chapters, we conducted a series of implementations and user studies to address these challenges. Our solutions involved mainly a combination of *human factors* and computational methods in 3DUIs. Our results suggest interesting insights on how to design more effective 3DUIs.

For instance, in [Chapter 3](#), we contribute a hybrid method to produce task-to-gesture mappings, combining both designers' insight (*high-level* factors) and computational approaches (*low-level* factors). The benefits of this hybrid approach become particularly clear for complex tasks with a large solution space (many potential mappings, as in our text-entry example). Our study showed a surprising outcome when the results of pure computational approaches are compared to those of our hybrid solution. All purely computational approaches predicted a very poor performance for the mapping that included designers' insight (i.e. our hybrid method). This could indicate that such insight could be irrelevant (or even harmful). This raised the question: "was the designer's insight misled or did computational approaches fail to capture the designer's insight?"

This question motivated our study and revealed that our hybrid solution actually produced very compelling results for all the factors measured. We strongly believe that these results, and the relatively low costs of using our method provide strong evidence of the potential benefits of the approach followed within this thesis. That is, they show how a combination of *human factors* and computational approaches can lead to the effective design of 3DUIs, even for a complex context (i.e., *high-level* factors, such as cognitive load, and a very large solution space).

Then in [Chapter 4](#), we leverage retargeting techniques for ergonomics purposes. Retargeting approaches have been used to create illusions in VR ([Gonzalez-Franco et al., 2017](#)) being beneficial for many applications in HCI (e.g., haptic feedback and navigation ([Slater et al., 1995](#); [Azmandian et al., 2016](#))). Considering that mid-air object *manipulation* is tiring ([Hincapié-Ramos et al., 2014](#)), we contribute Erg-O, a technique that uses online ergonomic evaluations to generate comfort mappings into the arms' reach (within the near interactive space) to enable object retargeting (re-mapping) aimed to reduce fatigue (i.e., gorilla arm syndrome ([Boring et al., 2009](#))) in object *manipulation* tasks. We highlight how the combination of retargeting and ergonomic evaluations can allow designers to modify users' perceptions to manipulate apparent-positions of virtual elements in the VE to achieve longer interaction time with a decreased perceived effort.

Our results also show that this ergonomic benefit can be achieved being unnoticed, and without significant effects on performance or sense of control. Also, our formalization and implementation demonstrate how these approaches can be seamlessly included in current manipulation techniques, even in real time, providing an example of how such interaction techniques can benefit from the computational power of current computers.

Following our exploration, in [Chapter 5](#) we then explored *scale-adaptive* navigation techniques to address space limitations involved when users' displacement in VR is achieved by actual physical locomotion. We gave special consideration on manoeuvring when a target is reached. Current solutions in the literature that apply *scale-adaptive* navigation are mainly focussed on users' perspective (*ego-centric*) to apply scaling factors and are not space-aware ([Feasel et al., 2008](#)). That is, scaling factors are also applied during manoeuvring tasks affecting precise exploration in relevant zones. To address this issue, we contribute NaviFields, a novel *allocentric* technique that accounts for the relevance of the space and applies scaling factors accordingly allowing to navigate bigger spaces along non-relevant areas while maintaining precise manoeuvring in relevant areas. Again, our approach illustrates how formal mathematical modelling and task awareness (e.g., the navigation field reflects the tasks that the user needs to complete in the VE) can lead to solutions providing better support for natural interaction (i.e., use physical displacement-based navigation across larger areas), while not hindering users' performance.

Along the same line of *navigation*, in [Chapter 6](#) we address the detrimental *Drift* effects induced by *scale-adaptive* navigation techniques. This issue causes a mismatch between the real/virtual spaces which can increase over time turning the navigation techniques unusable. To address this problem, we proposed a correction technique that dynamically applies correction factors according to the scaling policy of the *navigation* technique. Results from our studies show that our approach consistently reduced the *Drift* effect produced by two different techniques proposed by literature (one *ego-centric* and one *allocentric*), increasing thus their life-span (i.e., time that they can be used before *Drift* draws targets unreachable), while not hindering users' experience. Our results also suggest that, theoretically, any navigation technique using *scale-adaptive* factors can easily implement our correction approach as demonstrated in our user study. This chapter also highlights valuable aspects from our approach. First, it highlights the value of formal mathematical modelling, with the formalization of *Drift* in [Chapter 5](#) being key to allow

us to propose the solution described in [Chapter 6](#). Second, our solution requires simulations of the user returning to the centre VE to estimate Drift, which were run in real-time every frame. Our implementation shows how this is again within reach for current devices and illustrates promising benefits (i.e., correcting *Drift* for various environments and navigation techniques).

Finally, in [Chapter 7](#), we addressed the problem of scene density and occlusion in 3D *selection* tasks. *Selection* of 3D scanned models in VR (e.g., point clouds ~6M points) is extremely challenging due to inter-object occlusion and imprecise mid-air input (e.g., 6 DoF controllers). We then proposed a hybrid 2D-3D interaction technique that combines 3D selection in mid-air and 2D selection on a physical tablet. Inspired by well-known slicing plane techniques ([Hinckley et al., 1994](#)), our approach employs a slicing-volume that encloses 3D objects and maps them onto a 2D tablet view in VR, allowing users to select 3D points entirely on the tablet using 2D interactions. This technique advantages from both an extra visualization point (the tablet view), and haptic feedback (from the tangible tablet surface) giving stability while selecting. This technique made extensive use of compute shaders, to select, de-select and provide feedback to the user, which are only feasible by exploiting the computational power of current GPUs and which were key to explore this context of application (i.e., real-time interaction with highly dense point clouds). The results from our user study show that this technique can significantly improve the accuracy of selection in high dense models as it reduces under-selection and over-selection.

It is worth noting the independent contribution of this thesis on the exploration/identification of the advantages and limitations of isomorphic and non-isomorphic mapping generation, depending on the specific tasks and the *human factors* involved in the application, as demonstrated on [Chapters 4, 5 and 6](#).

## 8.2 Interaction Stages & General Contributions

In the previous section, we discussed the contributions of our approach under the frame of specific applications (local contribution in each chapter). In this section, we discuss the potential of the approach from a broader perspective based on the interaction stages proposed by ([Molina-Masso et al., 2008](#)) and shown in **Figure 2.4**. One of the main highlights from the results of the point studies, was that we explored the close coupling between specific applications and the ITes used, supporting the inclusion of *human factors*. For instance, in our exploration through interaction stages we found that:

Users performed a demanding task better ([Chapter 3](#)) when human's tacit knowledge was immersed in the process of the design and optimization of *input* mappings used for the task. This designer-led mapping proposed outperformed its competitors (a set of mappings generated by naive pure computational approaches). This highlights the benefits of our approach in the first stage of interaction (data *input*).

Similarly, we leverage *human factors* in ITes (the second stage on the interaction model) in [Chapter 4](#) and [Chapter 5](#). We explored ergonomics in object *manipulation* tasks as well as more natural *navigation* involving physical locomotion. Our results support the potential of including our approach on the generation of more tailored interplay between *human factors* and ITes to enhance naturalness of 3DUIs. While the approach in [Chapter 6](#) is not explicitly focused on considering *human factors*, it enables natural navigation techniques for extended periods of time which could be considered under this light.

In the third interaction stage explored (ITas), we showed that the combination between natural metaphors for VR selection tasks (i.e., bimanual interaction) that mimic users' interaction in the real world (e.g., *pen-and-tablet* metaphor); and the computational power from current hardware (GPU processing), lead to improvements on accuracy, stability and user experience (UX) (as shown in [Chapter 7](#)). We exploited human capabilities by combining 2D and 3D metaphors allowing bimanual interaction (unlike simple controller-based interaction in mid-air) while addressing limitations produced by inaccuracies of mid-air interaction (i.e., human ergonomic limitations when interacting at arbitrary 3D spaces).

It is worth mentioning that one of the goals of our general approach was not only to improve naturalness on the interaction with 3DUIs, but also to avoid disruptions on UX as it was observed on the results from each independent approach presented in this thesis, where we could improve ergonomics naturalness and accuracy without disrupting overall performance and sense of control.

We believe that with the findings and contributions obtained from this exploration (through the stages of interaction) we provide techniques and guidelines to design more natural 3DUIs involving gestural *input*, *navigation*, *selection* and object *manipulation* thanks to the combination of *human factors* and computational approaches using current hardware/software power. With this thesis, we then ambition to advance the knowledge to develop a standardization of 3DUIs comparable to that from traditional 2DUIs.

Moreover, one remarkable contribution of this thesis is the possible combination of the independent interaction stages we explored. We can illustrate the affordances of our general approach using three examples of how each independent approach presented here can be combined.

For instance, as discussed in [Chapter 5](#) and [Chapter 6](#), *Drift* effects increasingly disrupt the usability of the *navigation* technique, and therefore we proposed a correction approach in [Chapter 6](#) that significantly reduces this detrimental effect. However, we can combine (**example 1**): *Drift* correction techniques (to extend the time that the *navigation* technique is useful) with the use of isomorphism proposed in [Chapter 4](#) and [Chapter 5](#) (e.g., redirection and scaling factors) to extend also *navigation* spaces and the *endurance time* (i.e., the duration of user comfort during object *manipulation*), to support more complete VR interaction.

**Example 2:** The isomorphic mapping employed by Erg-O technique can also be employed to totally eliminate the *Drift* effect in *navigation*. For instance, by combining *navigation* fields ([Chapter 5](#)) and spatial partitioning ([Chapter 4](#)), we can subdivide the navigation plane in both worlds (real and virtual), generating equivalent subsections between spaces (each of those with an independent system of reference). This approach can allow us to generate a bijective mapping, where each sub-section in the real navigation plane will have its analogue in the virtual one. Once the virtual space is scaled, the analogy between sub-sections (in the virtual and real worlds) remains, enabling extended VR *navigation* without *Drift* effects. Then, specific sub-sections in the virtual navigation plane can be distorted (according to *navigation* fields approach) to support not only extended navigation but also allowing natural explorations in relevant zones while still tackling the *Drift* issue.

Our third example highlights the potential of our approach by combining the presented approaches in [Chapter 4](#) and [Chapter 7](#) (corresponding to object *manipulation* and *selection*). That is, we can combine (**example 3**): motion redirection in a smaller scale (e.g., pinch or tap gestures) for fine interaction in bimanual selection tasks (involving both hands) i.e., combining (i) retargeting techniques to influence users' perception; while (ii) exploiting bimanual metaphors for more natural interaction. Retargeting at lower scales can help to provide haptic feedback (e.g., by using proxies ([Slater et al., 1995](#); [Azmandian et al., 2016](#))) when manipulating small objects in VR, taking advantage of the potential of human hand dexterity for 3DUIs.



Our approach, combining *human factors*, mathematical modelling and computational approaches has allowed us to exploit human capabilities for interaction, changing in some manner the way “the system sees us” (**Figure 1.1**, right) (O’Sullivan et al., 2004) enriching the interaction and highlighting the need of more tailored ITes. Our exploration along the point studies in this work, suggests the potential of our approach and the need for more deep considerations when designing ITes and ITas. For instance, it is difficult to consider high-level factors (such as the case of [Chapter 3](#)) when using general formulation for 3DUI interactions (such as the case of usual ITes definitions), where a deeper exploration of the context around the specific task is needed to take advantage of the available hardware and methods to customize the interaction technique to better achieve the task.

The use of the mathematical formulations in our approach, allowed us first to understand the behaviours of the variables involved in the issue to be addressed, to better develop potential solutions. This initial understanding of the main variables led us to a more focused exploration to identify the set of elements (devices and techniques) that are more adequate to design an ITe tailored to each specific task, and therefore better address the issue as explored in this thesis.

### 8.3 Conclusion

With the solutions, studies and results presented in this thesis, we aim to contribute to the current research on 3D interaction. Designing 3DUIs is a challenging task considering that UIs are commonly standardized for 2D applications ([LaViola Jr et al., 2017](#)) and the limitations involved when interacting in 3D spaces (e.g., human ergonomic capabilities and hardware/software constraints). The need of new devices, techniques and metaphors to face the emerging requirements from 3DUIs, enables a challenging research to meet the usability requirements for design. Therefore, with the solutions presented along this thesis we aim to advance this research to design more effective and intuitive 3DUIs.

Moreover, the presented work in this thesis, not only highlights the need to improve current ITes design, but also aims to advance the understanding about the benefits of more exhaustive explorations of the interplay between *human factors*, ITes, ITas and devices, that can positively influence on the generation of more natural and intuitive 3DUI interactions.

## References

- Afonso, L., P. Dias, C. Ferreira and B. S. Santos (2017). Effect of hand-avatar in a selection task using a tablet as input device in an immersive virtual environment. 3D User Interfaces (3DUI), 2017 IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2017.7893364>.
- Agrawala, M., A. C. Beers and M. Levoy (1995). 3D painting on scanned surfaces. Proceedings of the 1995 symposium on Interactive 3D graphics, ACM. doi: <https://doi.org/10.1145/199404.199429>.
- Altahawi, F. and N. Subhas (2018). "3D MRI in Musculoskeletal Imaging: Current and Future Applications." *Current Radiology Reports* 6(8): 27. doi: <https://doi.org/10.1007/s40134-018-0287-3>.
- Anthes, C., P. Heinzlreiter, G. Kurka and J. Volkert (2004). Navigation models for a flexible, multi-mode VR navigation framework. Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry, ACM. doi: <https://doi.org/10.1145/1044588.1044693>.
- Argelaguet, F. (2014). Adaptive navigation for virtual environments. 3D User Interfaces (3DUI), 2014 IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2014.7027325>.
- Argelaguet, F. and C. Andujar (2013). "A survey of 3D object selection techniques for virtual environments." *Computers & Graphics* 37(3): 121-136. doi: <https://doi.org/10.1016/j.cag.2012.12.003>.
- Arora, R., I. Darolia, V. P. Namboodiri, K. Singh and A. Bousseau (2017). SketchSoup: Exploratory Ideation Using Design Sketches. *Computer Graphics Forum*, Wiley Online Library. doi: <https://doi.org/10.1111/cgf.13081>.
- Arora, R., R. Habib Kazi, T. Grossman, G. Fitzmaurice and K. Singh (2018). SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ACM. doi: <https://doi.org/10.1145/3173574.3173759>.
- Arora, R., R. H. Kazi, F. Anderson, T. Grossman, K. Singh and G. W. Fitzmaurice (2017). Experimental Evaluation of Sketching on Surfaces in VR. CHI. doi: <http://dx.doi.org/10.1145/3025453.3025474>.
- Aspin, R. and K. H. Le (2007). Augmenting the CAVE: An initial study into close focused, inward looking, exploration in IPT systems. 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'07), IEEE. doi: <https://doi.org/10.1109/DS-RT.2007.9>.
- Azmadian, M., T. Grechkin, M. T. Bolas and E. A. Suma (2015). Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance. ICAT-EGVE. doi: <http://dx.doi.org/10.2312/egve.20151315>.
- Azmadian, M., M. Hancock, H. Benko, E. Ofek and A. D. Wilson (2016). Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM. doi: <http://dx.doi.org/10.1145/2858036.2858226>.
- Bachynskyi, M., G. Palmas, A. Oulasvirta and T. Weinkauff (2015). "Informing the design of novel input methods with muscle coactivation clustering." *ACM Transactions on Computer-Human Interaction (TOCHI)* 21(6): 30. doi: <http://dx.doi.org/10.1145/2687921>.

- Bacim, F., R. Kopper and D. A. Bowman (2013). "Design and evaluation of 3D selection techniques based on progressive refinement." *International Journal of Human-Computer Studies* **71**(7-8): 785-802. doi: <https://doi.org/10.1016/j.ijhcs.2013.03.003>.
- Badler, N. I., C. B. Phillips and B. L. Webber (1993). *Simulating humans: computer graphics animation and control*, Oxford University Press.
- Ban, Y., T. Kajinami, T. Narumi, T. Tanikawa and M. Hirose (2012). Modifying an identified curved surface shape using pseudo-haptic effect. *Haptics Symposium (HAPTICS)*, 2012 IEEE, IEEE. doi: [http://dx.doi.org/10.1007/978-3-642-31401-8\\_3](http://dx.doi.org/10.1007/978-3-642-31401-8_3).
- Banakou, D., R. Groten and M. Slater (2013). "Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes." *Proceedings of the National Academy of Sciences* **110**(31): 12846-12851. doi: <http://dx.doi.org/10.1073/pnas.1306779110>.
- Banakou, D. and M. Slater (2014). "Body ownership causes illusory self-attribution of speaking and influences subsequent real speaking." *Proceedings of the National Academy of Sciences* **111**(49): 17678-17683. doi: <http://dx.doi.org/10.1073/pnas.1414936111>.
- Barbagli, F., K. Salisbury, C. Ho, C. Spence and H. Z. Tan (2006). "Haptic discrimination of force direction and the influence of visual information." *ACM Transactions on Applied Perception (TAP)* **3**(2): 125-135. doi: <http://dx.doi.org/10.1145/1141897.1141901>.
- Barrilleaux, J. (2001). *3D user interfaces with Java 3D*, Manning Greenwich.
- Benko, H. and S. Feiner (2007). Balloon selection: A multi-finger technique for accurate low-fatigue 3d selection. *2007 IEEE Symposium on 3D User Interfaces*, IEEE. doi: <https://doi.org/10.1109/3DUI.2007.340778>.
- Besançon, L., P. Issartel, M. Ammi and T. Isenberg (2017). "Hybrid tactile/tangible interaction for 3D data exploration." *IEEE transactions on visualization and computer graphics* **23**(1): 881-890. doi: <https://doi.org/10.1109/TVCG.2016.2599217>.
- Bhaskar Surale, H., A. Gupta, M. Hancock and D. Vogel (2019). *TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality*. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Glasgow, Scotland, ACM. doi: <https://doi.org/10.1145/3290605.3300243>.
- Bi, X., B. A. Smith and S. Zhai (2010). Quasi-qwerty soft keyboard optimization. *Proceedings of the SIGCHI Conference*, ACM. doi: <https://doi.org/10.1145/1753326.1753367>.
- Bi, X. and S. Zhai (2016). IJQwerty: What Difference Does One Key Change Make? *Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty*. *Proceedings of the 2016 CHI Conference*, ACM. doi: <https://doi.org/10.1145/2858036.2858421>.
- Billinghurst, M., S. Baldis, L. Matheson and M. Philips (1997). 3D palette: a virtual reality content creation tool. *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM. doi: <https://doi.org/10.1145/261135.261163>.
- Borg, G. (1990). "Psychophysical scaling with applications in physical work and the perception of exertion." *Scandinavian journal of work, environment & health*: 55-58. doi: <https://doi.org/10.5271/sjweh.1815>.
- Borg, G. (1998). *Borg's perceived exertion and pain scales*, Human kinetics.

- Boring, S., M. Jurmu and A. Butz (2009). Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7, ACM. doi: <http://dx.doi.org/10.1145/1738826.1738853>.
- Bornik, A., R. Beichel, E. Kruijff, B. Reitinger and D. Schmalstieg (2006). A hybrid user interface for manipulation of volumetric medical data. null, IEEE. doi: <http://doi.ieeecomputersociety.org/10.1109/VR.2006.8>.
- Bowman, D., E. Kruijff, J. J. LaViola Jr and I. P. Poupyrev (2004). 3D User interfaces: theory and practice, CourseSmart eTextbook, Addison-Wesley.
- Bowman, D. A., J. L. Gabbard and D. Hix (2002). "A survey of usability evaluation in virtual environments: classification and comparison of methods." Presence: Teleoperators & Virtual Environments **11**(4): 404-424. doi: <https://doi.org/10.1162/105474602760204309>
- Bowman, D. A. and L. F. Hodges (1997). An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. Proceedings of the 1997 symposium on Interactive 3D graphics, ACM. doi: <http://dx.doi.org/10.1145/253284.253301>.
- Bowman, D. A. and L. F. Hodges (1999). "Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments." Journal of Visual Languages & Computing **10**(1): 37-53. doi: <https://doi.org/10.1006/jvlc.1998.0111>.
- Bowman, D. A., D. Koller and L. F. Hodges (1997). Travel in immersive virtual environments: an evaluation of viewpoint motion control techniques. Proceedings of IEEE 1997 Annual International Symposium on Virtual Reality. doi: <https://doi.org/10.1109/VRAIS.1997.583043>.
- Bowman, D. A., E. Kruijff, J. J. LaViola and I. Poupyrev (2004). 3D User Interfaces: Theory and Practice, Addison Wesley Longman Publishing Co., Inc.,
- Bowman, D. A., E. Kruijff, J. J. LaViola Jr and I. Poupyrev (2001). "An Introduction to 3-D User Interface Design." Presence: Teleoperators and virtual environments **10**(1): 96-108. doi: <https://doi.org/10.1162/105474601750182342>.
- Bowman, D. A., M. Setareh, M. S. Pinho, N. Ali, A. Kalita, Y. Lee, J. Lucas, M. Gracey, M. Kothapalli and Q. Zhu (2003). Virtual-SAP: an immersive tool for visualizing the response of building structures to environmental conditions. IEEE Virtual Reality, 2003. Proceedings., IEEE. doi: <https://doi.org/10.1109/VR.2003.1191146>.
- Bowman, D. A. and C. A. Wingrave (2001). Design and evaluation of menu systems for immersive virtual environments. Proceedings IEEE Virtual Reality 2001, IEEE. doi: <https://doi.org/10.1109/VR.2001.913781>.
- Boyd, D. and L. Sastry (1999). Development of the INQUISITIVE Interaction Toolkit-Concept and Realisation. Proc. of Workshop on User Centered Design and Implementation of Virtual Environments UC DIVE'99.
- Bozgeyikli, E., A. Raij, S. Katkoori and R. Dubey (2016). Point & teleport locomotion technique for virtual reality. Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play, ACM. doi: <https://doi.org/10.1145/2967934.2968105>.

- Bricken, M. (1991). "Virtual reality learning environments: potentials and challenges." ACM SIGGRAPH Computer Graphics **25**(3): 178-184. doi: <http://dx.doi.org/10.1145/126640.126657>.
- Brown University (2014). "Virtual Lab: Map your homunculus." Available from: [https://canvas.brown.edu/courses/851434/assignments/4953274?module\\_item\\_id=6841193](https://canvas.brown.edu/courses/851434/assignments/4953274?module_item_id=6841193).
- Burdorf, A. and J. Laan (1991). "Comparison of methods for the assessment of postural load on the back." Scandinavian journal of work, environment & health: 425-429. doi: <https://doi.org/10.5271/sjweh.1679>.
- Burns, E., S. Razzaque, A. T. Panter, M. C. Whitton, M. R. McCallus and F. P. Brooks (2005). The hand is slower than the eye: A quantitative exploration of visual dominance over proprioception. Virtual Reality, 2005. Proceedings. VR 2005. IEEE, IEEE. doi: <http://dx.doi.org/10.1109/VR.2005.1492747>.
- Bustamante, E. A. and R. D. Spain (2008). Measurement invariance of the Nasa TLX. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, SAGE Publications Sage CA: Los Angeles, CA. doi: <http://dx.doi.org/10.1177/154193120805201946>.
- Bystrom, K.-E., W. Barfield and C. Hendrix (1999). "A conceptual model of the sense of presence in virtual environments." Presence: Teleoperators and virtual environments **8**(2): 241-244. doi: <http://dx.doi.org/10.1162/105474699566107>.
- Cadoz, C. (1994). Les réalités virtuelles.
- Chaffin, D. B., G. Andersson and B. J. Martin (1999). Occupational biomechanics, Wiley New York.
- Cheng, L.-P., E. Ofek, C. Holz, H. Benko and A. D. Wilson (2017). Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, ACM. doi: <https://doi.org/10.1145/3025453.3025753>.
- Cheng, Y.-L. and K. S. Mix (2014). "Spatial training improves children's mathematics ability." Journal of Cognition and Development **15**(1): 2-11. doi: <http://dx.doi.org/10.1080/15248372.2012.725186>.
- Cheveau, A. (2018). 3D Scanning Solution for Textured Object using Photometric Stereo with Multiple Known Light Sources. Archiving Conference, Society for Imaging Science and Technology. doi: <https://doi.org/10.2352/issn.2168-3204.2018.1.0.3>.
- Cirio, G., M. Marchal, T. Regia-Corte and A. Lécuyer (2009). The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace. ACM Symposium on Virtual Reality Software and Technology, ACM. doi: <https://doi.org/10.1145/1643928.1643965>.
- Cockburn, A. and A. Firth (2004). Improving the acquisition of small targets. People and Computers XVII—Designing for Society, Springer: 181-196. doi: [https://doi.org/10.1007/978-1-4471-3754-2\\_11](https://doi.org/10.1007/978-1-4471-3754-2_11).
- Cohen, J. (1988). "Statistical power analysis for the behavioral sciences . Hillsdale." NJ: Lawrence Earlbaum Associates **2**. <https://www.taylorfrancis.com/books/9781134742707>
- Colman, A. M. (2015). A dictionary of psychology, Oxford University Press, USA.
- Community, S. (2016). "SteamVR Play Area Size Stats." Available from: <https://bit.ly/2HzlvpG>.
- Community, S. (2017). "VR Roomscale room size survey - answers analysis." Available from: <https://bit.ly/2U7yGDW>.

- Conner, B. D., S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik and A. Van Dam (1992). Three-dimensional widgets. Proceedings of the 1992 symposium on Interactive 3D graphics, ACM. doi: <https://doi.org/10.1145/147156.147199>.
- Cruz-Neira, C., J. Leigh, M. Papka, C. Barnes, S. M. Cohen, S. Das, R. Engelmann, R. Hudson, T. Roy and L. Siegel (1993). Scientists in wonderland: A report on visualization applications in the CAVE virtual reality environment. Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium, IEEE. doi: <https://doi.org/10.1109/VRAIS.1993.378262>.
- Cumming, G., F. Fidler and D. L. Vaux (2007). "Error bars in experimental biology." The Journal of cell biology **177**(1): 7-11. doi: <https://doi.org/10.1083/jcb.200611141>.
- Cutting, J. E. (1997). "How the eye measures reality and virtual reality." Behavior Research Methods **29**(1): 27-36. doi: <https://doi.org/10.3758/BF03200563>.
- Dachselt, R. and M. Hinz (2005). Three-dimensional widgets revisited-towards future standardization. IEEE VR 2005 Workshop New directions in 3D user interfaces.
- Darken, R. P., W. R. Cockayne and D. Carmein (1997). The omni-directional treadmill: a locomotion device for virtual worlds. Proceedings of the 10th annual ACM symposium on User interface software and technology, ACM. doi: <https://doi.org/10.1145/263407.263550>.
- Darken, R. P., W. R. Cockayne and D. Carmein (1997). The omni-directional treadmill: a locomotion device for virtual worlds. ACM symposium on User interface software and technology, ACM. doi: <https://doi.org/10.1145/263407.263550>.
- Darken, R. P. and J. L. Sibert (1993). A toolset for navigation in virtual environments. ACM symposium on User interface software and technology, ACM. doi: <https://doi.org/10.1145/168642.168658>.
- Davey, B. and K. R. Parker (2015). "Requirements elicitation problems: a literature analysis." Issues in Informing Science and Information Technology **12**: 71-82. doi: <https://doi.org/10.28945/2211>.
- David, R. (1998). "RealPlaces, 3D interface for office applications." IEE colloquium on the 3D Interface for the information Worker '98: 437. doi: <http://www3.ibm.com/ibm/easy/eouext.nsf/Publish/580>.
- de Haan, G., J. Scheuer, R. de Vries and F. H. Post (2009). Egocentric navigation for video surveillance in 3D virtual environments. 3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2009.4811214>.
- Dias, P., L. Afonso, S. Eliseu and B. S. Santos (2018). Mobile devices for interaction in immersive virtual environments. Proceedings of the 2018 International Conference on Advanced Visual Interfaces, ACM. doi: <https://doi.org/10.1145/3206505.3206526>.
- Dunne, A., S. Do-Lenh, G. Ó'Laighin, C. Shen and P. Bonato (2010). Upper extremity rehabilitation of children with cerebral palsy using accelerometer feedback on a multitouch display. Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE. doi: <http://dx.doi.org/10.1109/IEMBS.2010.5626724>.
- Dünser, A., K. Steinbügl, H. Kaufmann and J. Glück (2006). Virtual and augmented reality as spatial ability training tools. Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centered HCI, ACM. doi: <http://dx.doi.org/10.1145/1152760.1152776>.



- Dutta, T. (2012). "Evaluation of the Kinect™ sensor for 3-D kinematic measurement in the workplace." *Applied ergonomics* **43**(4): 645-649. doi: <https://doi.org/10.1016/j.apergo.2011.09.011>.
- Eastgate, R. M. (2001). *The structured development of virtual environments: enhancing functionality and interactivity*, University of Nottingham. doi: <http://eprints.nottingham.ac.uk/id/eprint/10954>.
- Fabiani, L., G. C. Burdea, N. A. Langrana and D. Gomez (1996). Human interface using the Rutgers Master II force feedback interface. VRAIS. doi: <https://doi.org/10.1109/VRAIS.1996.490510>.
- Feasel, J., M. C. Whitton and J. D. Wendt (2008). LLCM-WIP: Low-latency, continuous-motion walking-in-place. 3D User Interfaces, 2008. 3DUI 2008. IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2008.4476598>.
- Feit, A. M. and A. Oulasvirta (2014). Pianotext: Redesigning the piano keyboard for text entry. *Proceedings of the 2014 conference DIS*, ACM. doi: <https://doi.org/10.1145/2598510.2598547>.
- Fernandes, K. J., V. Raja and J. Eyre (2003). "Cybersphere: the fully immersive spherical projection system." *Communications of the ACM* **46**(9): 141-146. doi: <https://doi.org/10.1145/903893.903929>.
- Flasar, J. (2001). *Interaction Techniques for Object Selection/Manipulation in Non-Immersive Virtual Environments with Force Feedback*, Eurohaptics. URL: <http://www.eurohaptics.vision.ee.ethz.ch/2001/flasar.pdf>
- Foley, J., A. Van Dam, J. Feiner and J. Hughes (1990). *Computer Graphics, Principles and Practice*, 2nd (ed.,) Addison-Wesley.
- Foley, J. D., V. L. Wallace and P. Chan (1984). "The human factors of computer graphics interaction techniques." *IEEE computer Graphics and Applications* **4**(11): 13-48. doi: <https://doi.org/10.1109/MCG.1984.6429355>.
- Forsberg, A., K. Herndon and R. Zeleznik (1996). Aperture based selection for immersive virtual environments. *ACM Symposium on User Interface Software and Technology*, Citeseer. doi: <https://doi.org/10.1145/237091.237105>.
- Foxlin, E. (2005). "Pedestrian tracking with shoe-mounted inertial sensors." *IEEE computer Graphics and Applications* **25**(6): 38-46. doi: <https://doi.org/10.1109/MCG.2005.140>.
- Freitag, S., B. Weyers and T. W. Kuhlen (2016). Automatic speed adjustment for travel through immersive virtual environments based on viewpoint quality. 3D User Interfaces (3DUI), 2016 IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2016.7460033>.
- Fuhrmann, A., D. Schmalstieg and M. Gervautz (1998). Strolling through cyberspace with your hands in your pockets: Head directed navigation in virtual environments. *Virtual Environments '98*, Springer: 216-225. doi: [https://doi.org/10.1007/978-3-7091-7519-4\\_21](https://doi.org/10.1007/978-3-7091-7519-4_21).
- Gabbard, J. L., D. Hix and J. E. Swan (1999). "User-centered design and evaluation of virtual environments." *IEEE computer Graphics and Applications* **19**(6): 51-59. doi: <https://doi.org/10.1109/38.799740>.
- García, A., J. Molina and P. González (2005). Exemplar VE design guidance tool for selection and manipulation interaction techniques. *Proc. of 11th International Conference on Human-Computer Interaction*, HCI International.

- Gelain, M., M. S. Pini, F. Rossi, K. B. Venable and T. Walsh (2010). "Elicitation strategies for soft constraint problems with missing preferences: Properties, algorithms and experimental studies." *Artificial Intelligence* **174**(3-4): 270-294. doi: <https://doi.org/10.1016/j.artint.2009.11.015>.
- Gelbart, D. and S. R. Zmtiule (1968). "A research center for augmenting human intellect." doi: <https://doi.org/10.1145/1476589.1476645>.
- Giesler, A., D. Valkov and K. Hinrichs (2014). Void shadows: multi-touch interaction with stereoscopic objects on the tabletop. *Proceedings of the 2nd ACM symposium on Spatial user interaction*, ACM. doi: <https://doi.org/10.1145/2659766.2659779>.
- Gong, J., B. Haggerty and P. Tarasewich (2005). An enhanced multitap text entry method with predictive next-letter highlighting. *CHI'05*, ACM. doi: <https://doi.org/10.1145/1056808.1056926>.
- González, G., J. P. Molina, A. S. García, D. Martínez and P. González (2009). Evaluation of text input techniques in immersive virtual environments. *New Trends on Human-Computer Interaction*, Springer: 109-118. doi: [https://doi.org/10.1007/978-1-84882-352-5\\_11](https://doi.org/10.1007/978-1-84882-352-5_11).
- Gonzalez-Franco, M. and J. Lanier (2017). "Model of illusions and virtual reality." *Frontiers in psychology* **8**: 1125. doi: <https://doi.org/10.3389/fpsyg.2017.01125>.
- Grossman, T. and R. Balakrishnan (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM. doi: <https://doi.org/10.1145/1054972.1055012>.
- Grossman, T. and R. Balakrishnan (2006). The design and evaluation of selection techniques for 3D volumetric displays. *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM. doi: <https://doi.org/10.1145/1166253.1166257>.
- Grossman, T., D. Wigdor and R. Balakrishnan (2004). Multi-finger gestural interaction with 3d volumetric displays. *Proceedings of the 17th annual ACM UIST*, ACM. doi: <https://doi.org/10.1145/1029632.1029644>.
- Haldar, R. and D. Mukhopadhyay (2011). "Levenshtein distance technique in dictionary lookup methods: An improved approach." *arXiv e-print (arXiv:1101.1232)*. doi: [https://doi.org/10.1007/978-981-13-0755-3\\_6](https://doi.org/10.1007/978-981-13-0755-3_6).
- Hale, K. S. and K. M. Stanney (2014). *Handbook of virtual environments: Design, implementation, and applications*, CRC Press.
- Harris, L., M. Jenkin and D. C. Zikovitz (1999). Vestibular cues and virtual environments: choosing the magnitude of the vestibular cue. *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, IEEE. doi: <https://doi.org/10.1109/VR.1999.756956>.
- Hayhoe, M., B. Gillam, K. Chajka and E. Vecellio (2009). "The role of binocular vision in walking." *Visual neuroscience* **26**(1): 73-80. doi: <https://doi.org/10.1017/S0952523808080838>.
- Herndon, K. P., A. van Dam and M. Gleicher (1994). "The challenges of 3D interaction: a CHI'94 workshop." *ACM SIGCHI Bulletin* **26**(4): 36-43. doi: <https://doi.org/10.1145/191642.191652>.
- Hincapié-Ramos, J. D., X. Guo, P. Moghadasian and P. Irani (2014). Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, ACM. doi: <http://dx.doi.org/10.1145/2556288.2557130>.



- Hinckley, K., R. Pausch, J. C. Goble and N. F. Kassell (1994). Passive real-world interface props for neurosurgical visualization. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Boston, Massachusetts, USA, ACM: 452-458. doi: <https://doi.org/10.1145/191666.191821>.
- Hix, D. and H. R. Hartson (1993). Developing user interfaces: ensuring usability through product & process, John Wiley & Sons, Inc.,
- Horn, R. A., R. A. Horn and C. R. Johnson (1990). Matrix analysis, Cambridge university press.
- Interrante, V., B. Ries and L. Anderson (2007). Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. 3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on, IEEE. doi: <https://doi.org/10.1109/3DUI.2007.340791>.
- Izadi, S., D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman and A. Davison (2011). KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. Proceedings of the 24th annual ACM symposium on User interface software and technology, ACM. doi: <https://doi.org/10.1145/2047196.2047270>.
- Jackson, B., B. Jelke and G. Brown (2018). Yea Big, Yea High: A 3D User Interface for Surface Selection by Progressive Refinement in Virtual Environments. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), IEEE. doi: <https://doi.org/10.1109/VR.2018.8447559>.
- Janoos, F., B. Nouanesengsy, R. Machiraju, H. W. Shen, S. Sammet, M. Knopp and I. Á. Mórocz (2009). Visual analysis of brain activity from fMRI data. Computer Graphics Forum, Wiley Online Library. doi: <https://doi.org/10.1111/j.1467-8659.2009.01458.x>.
- Jones, L. A. and S. J. Lederman (2006). Human hand function, Oxford University Press.
- Kammer, D., J. Wojdziak, M. Keck, R. Groh and S. Taranko (2010). Towards a formalization of multi-touch gestures. ACM ISS, ACM: 49-58. doi: <https://doi.org/10.1145/1936652.1936662>.
- Kawato, M. (1999). "Internal models for motor control and trajectory planning." Current opinion in neurobiology 9(6): 718-727. doi: [https://doi.org/10.1016/S0959-4388\(99\)00028-8](https://doi.org/10.1016/S0959-4388(99)00028-8).
- Keefe, D. F., R. C. Zeleznik and D. H. Laidlaw (2007). "Drawing on air: Input techniques for controlled 3D line illustration." IEEE transactions on visualization and computer graphics 13(5): 1067-1081. doi: <https://doi.org/10.1109/TVCG.2007.1060>.
- Kerber, F., P. Lessel, A. Kr, #252 and ger (2015). Same-side Hand Interactions with Arm-placed Devices Using EMG. Proceedings of the ACM CHI. Seoul, Republic of Korea, ACM: 1367-1372. doi: <https://doi.org/10.1145/2702613.2732895>.
- Kessler, G. D., L. F. Hodges and N. Walker (1995). "Evaluation of the CyberGlove as a whole-hand input device." ACM TOCHI'95 2(4): 263-283. doi: <https://doi.org/10.1145/212430.212431>.
- Kilteni, K., J.-M. Normand, M. V. Sanchez-Vives and M. Slater (2012). "Extending body space in immersive virtual reality: a very long arm illusion." PloS one 7(7): e40867. doi: <http://dx.doi.org/10.1371/journal.pone.0040867>.
- Kim, D., O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis and P. Olivier (2012). Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. Proceedings of ACM UIST, ACM. doi: <https://doi.org/10.1145/2380116.2380139>.

- Kim, J.-S., D. Gračanin, K. Matković and F. Quek (2008). Finger walking in place (FWIP): A traveling technique in virtual environments. *International Symposium on Smart Graphics*, Springer. doi: [https://doi.org/10.1007/978-3-540-85412-8\\_6](https://doi.org/10.1007/978-3-540-85412-8_6).
- Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983). "Optimization by simulated annealing." *science* **220**(4598): 671-680. doi: <https://doi.org/10.1126/science.220.4598.671>.
- Klimt, B. and Y. Yang (2004). The enron corpus: A new dataset for email classification research. *European Conference on Machine Learning*, Springer. doi: [https://doi.org/10.1007/978-3-540-30115-8\\_22](https://doi.org/10.1007/978-3-540-30115-8_22).
- Knibbe, J., D. Martinez Plasencia, C. Bainbridge, C.-K. Chan, J. Wu, T. Cable, H. Munir and D. Coyle (2014). Extending interaction for smart watches: enabling bimanual around device control. *CHI'14*, ACM. doi: <https://doi.org/10.1145/2559206.2581315>.
- Knight, J. L. (1987). "Manual control and tracking." *Handbook of human factors*: 182-218.
- Kohli, L. (2010). Redirected touching: Warping space to remap passive haptics. *3D User Interfaces (3DUI)*, 2010 IEEE Symposium on, IEEE. doi: <http://dx.doi.org/10.1109/3DUI.2010.5444703>.
- Kohli, L., M. C. Whitton and F. P. Brooks (2012). Redirected touching: The effect of warping space on task performance. *3D User Interfaces (3DUI)*, 2012 IEEE Symposium on, IEEE. doi: <http://dx.doi.org/10.1109/3DUI.2012.6184193>.
- Kokkinara, E., K. Kiltner, K. J. Blom and M. Slater (2016). "First Person Perspective of Seated Participants Over a Walking Virtual Body Leads to Illusory Agency Over the Walking." *Scientific Reports* **6**. doi: <http://dx.doi.org/10.1038/srep28879>.
- Kopper, R., F. Bacim and D. A. Bowman (2011). Rapid and accurate 3D selection by progressive refinement. *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, IEEE. doi: <https://doi.org/10.1109/3DUI.2011.5759219>.
- Kopper, R., T. Ni, D. A. Bowman and M. Pinho (2006). Design and evaluation of navigation techniques for multiscale virtual environments. *Virtual Reality Conference, 2006*, Ieee. doi: <https://doi.org/10.1109/VR.2006.47>.
- Krueger, M. W., T. Gionfriddo and K. Hinrichsen (1985). Videoplace—an artificial reality.—*CHI'85: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press. doi: <https://doi.org/10.1145/317456.317463>.
- Lathrop, W. B. and M. K. Kaiser (2002). "Perceived orientation in physical and virtual environments: changes in perceived orientation as a function of idiothetic information available." *Presence* **11**(1): 19-32. doi: <https://doi.org/10.1162/105474602317343631>.
- Laurel, B., R. Strickland and R. Tow (1994). "Placeholder: Landscape and narrative in virtual environments." *ACM SIGGRAPH Computer Graphics* **28**(2): 118-126. doi: <https://doi.org/10.1145/178951.178967>.
- LaViola, J. J. (2013). "3d gestural interaction: The state of the field." *International Scholarly Research Notices* **2013**. doi: <http://dx.doi.org/10.1155/2013/514641>.
- LaViola Jr, J. J., D. A. Feliz, D. F. Keefe and R. C. Zeleznik (2001). Hands-free multi-scale navigation in virtual environments. *Symposium on Interactive 3D graphics*, ACM. doi: <https://doi.org/10.1145/364338.364339>.

- LaViola Jr, J. J., E. Kruijff, R. P. McMahan, D. Bowman and I. P. Poupyrev (2017). 3D user interfaces: theory and practice, Addison-Wesley Professional.
- Lécuyer, A., J.-M. Burkhardt, J.-M. Henaff and S. Donikian (2006). Camera motions improve the sensation of walking in virtual environments. Virtual Reality Conference, IEEE. doi: <https://doi.org/10.1109/VR.2006.31>.
- Lee, Y., I. Jang and D. Lee (2015). Enlarging just noticeable differences of visual-proprioceptive conflict in VR using haptic feedback. World Haptics Conference (WHC), 2015 IEEE, IEEE. doi: <http://dx.doi.org/10.1109/WHC.2015.7177685>.
- Lindeman, R. W. (1999). Bimanual interaction, passive-haptic feedback, 3 D widget representation, and simulated surface constraints for interaction in immersive virtual environments, George Washington University. [https://web.cs.wpi.edu/~gogo/papers/lindeman\\_thesis.pdf](https://web.cs.wpi.edu/~gogo/papers/lindeman_thesis.pdf)
- Liu, J., H. Parekh, M. Al-Zayer and E. Folmer (2018). Increasing Walking in VR using Redirected Teleportation. The 31st Annual ACM Symposium on User Interface Software and Technology, ACM. doi: <https://doi.org/10.1145/3242587.3242601>.
- Lopez, C., P. Halje and O. Blanke (2008). "Body ownership and embodiment: vestibular and multisensory mechanisms." *Neurophysiologie Clinique/Clinical Neurophysiology* **38**(3): 149-161. doi: <http://dx.doi.org/10.1016/j.neucli.2007.12.006>.
- López, L. B. (2013). Travel simulation inside an Immersive Video Environment (IVE). Sitcom Lab, Institut für Geoinformatik (IFGI) of the Westfälische Wilhelms-Universität Münster **MSc**. <http://hdl.handle.net/10362/9192>
- Lucas, J. F. (2005). Design and evaluation of 3D multiple object selection techniques, Virginia Tech. <http://hdl.handle.net/10919/31769>
- Mackinlay, J. D., S. K. Card and G. G. Robertson (1990). Rapid controlled movement through a virtual 3D workspace. ACM SIGGRAPH Computer Graphics, ACM. doi: <https://doi.org/10.1145/97880.97898>.
- Marsh, T., P. Wright, S. Smith and D. Duke (1998). "A shared framework of virtual reality." *Proceedings of UK-VRSIG* **98**.
- Martinez-Plasencia, D. (2010). Afreeca. An architecture for the development of collaborative virtual environment, Universidad de Castilla-La Mancha. <https://dialnet.unirioja.es/servlet/tesis?codigo=123978>
- Matsuoka, Y., S. J. Allin and R. L. Klatzky (2002). "The tolerance for visual feedback distortions in a virtual environment." *Physiology & behavior* **77**(4): 651-655. doi: [http://dx.doi.org/10.1016/S0031-9384\(02\)00914-9](http://dx.doi.org/10.1016/S0031-9384(02)00914-9).
- McAtamney, L. and E. N. Corlett (1993). "RULA: a survey method for the investigation of work-related upper limb disorders." *Applied ergonomics* **24**(2): 91-99. doi: [http://dx.doi.org/10.1016/0003-6870\(93\)90080-S](http://dx.doi.org/10.1016/0003-6870(93)90080-S).
- McIntosh, J., C. McNeill, M. Fraser, F. Kerber, M. Löchtefeld and A. Krüger (2016). EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. Proceedings of the 2016 CHI Conference, ACM. doi: <https://doi.org/10.1145/2858036.2858093>.

- McKeown, M. J., L. K. Hansen and T. J. Sejnowski (2003). "Independent component analysis of functional MRI: what is signal and what is noise?" *Current opinion in neurobiology* **13**(5): 620-629. doi: <https://doi.org/10.1016/j.conb.2003.09.012>.
- McNeill, D. (1992). *Hand and mind: What gestures reveal about thought*, University of Chicago press.
- Medeiros, D., F. Carvalho, L. Teixeira, P. Braz, A. Raposo and I. Santos (2013). "Proposal and evaluation of a tablet-based tool for 3D virtual environments." *SBC Journal on Interactive Systems* **4**(2): 31.
- Medeiros, D., L. Teixeira, F. Carvalho, I. Santos and A. Raposo (2013). A tablet-based 3d interaction tool for virtual engineering environments. *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ACM. doi: <https://doi.org/10.1145/2534329.2534349>.
- Medina, E., R. Fruland and S. Weghorst (2008). *Virtusphere: Walking in a human size VR "hamster ball"*. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA. doi: <https://doi.org/10.1177/154193120805202704>.
- Mendes, D., D. Medeiros, M. Sousa, E. Cordeiro, A. Ferreira and J. A. Jorge (2017). "Design and evaluation of a novel out-of-reach selection technique for VR using iterative refinement." *Computers & Graphics* **67**: 95-102. doi: <https://doi.org/10.1016/j.cag.2017.06.003>.
- Mine, M. R. (1995). "Virtual environment interaction techniques." UNC Chapel Hill CS Dept.
- Molina, J. P., P. González, M. D. Lozano, F. Montero and V. López-Jaquero (2003). *Bridging the gap: developing 2D and 3D user interfaces with the IDEAS methodology*. *International Workshop on Design, Specification, and Verification of Interactive Systems*, Springer. doi: [https://doi.org/10.1007/978-3-540-39929-2\\_21](https://doi.org/10.1007/978-3-540-39929-2_21).
- Molina-Masso, J. P. and P. G. Lopez (2008). "Un enfoque estructurado para el desarrollo de interfaces de usuario 3d." *Def. Doutorado-Universida de Castilla de la Mancha*. <https://dialnet.unirioja.es/servlet/tesis?codigo=21060>
- Mossel, A. and C. Koessler (2016). Large scale cut plane: an occlusion management technique for immersive dense 3D reconstructions. *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, ACM. doi: <https://doi.org/10.1145/2993369.2993384>.
- Mossel, A., B. Venditti and H. Kaufmann (2013). DrillSample: precise selection in dense handheld augmented reality environments. *Proceedings of the Virtual Reality International Conference: Laval Virtual*, ACM. doi: <https://doi.org/10.1145/2466816.2466827>.
- Nilsson, N. C., S. Serafin and R. Nordahl (2013). The perceived naturalness of virtual locomotion methods devoid of explicit leg movements. *Proceedings of Motion on Games*, ACM. doi: <https://doi.org/10.1145/2522628.2522655>.
- Normand, J.-M., E. Giannopoulos, B. Spanlang and M. Slater (2011). "Multisensory stimulation can induce an illusion of larger belly size in immersive virtual reality." *PloS one* **6**(1): e16128. doi: <http://dx.doi.org/10.1371/journal.pone.0016128>.
- O'Sullivan, D. and T. Igoe (2004). *Physical computing: sensing and controlling the physical world with computers*, Course Technology Press.

- Obermaier, H. and K. I. Joy (2015). "An Automated Approach for Slicing Plane Placement in Visual Data Analysis." *IEEE transactions on visualization and computer graphics* **21**(12): 1403-1414. doi: <https://doi.org/10.1109/TVCG.2015.2414455>.
- Olwal, A. and S. Feiner (2003). The flexible pointer: An interaction technique for selection in augmented and virtual reality. *Proc. UIST'03*.
- Oskiper, T., H.-P. Chiu, Z. Zhu, S. Samaresekera and R. Kumar (2011). Stable vision-aided navigation for large-area augmented reality. *Virtual Reality Conference (VR), 2011 IEEE, IEEE*. doi: <https://doi.org/10.1109/VR.2011.5759438>.
- Ott, M. and L. FREINA (2015). A literature review on immersive virtual reality in education: state of the art and perspectives. *Conference proceedings of eLearning and Software for Education «(eLSE)*, Universitatea Nationala de Aparare Carol I. URL: <https://www.cceol.com/search/article-detail?id=289829>
- Pavlovych, A. and W. Stuerzlinger (2004). Model for non-expert text entry speed on 12-button phone keypads. *Proceedings of the SIGCHI conference, ACM*. doi: <https://doi.org/10.1145/985692.985737>.
- Peck, T. C., H. Fuchs and M. C. Whitton (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: <https://doi.org/10.1109/TVCG.2008.191>.
- Peck, T. C., H. Fuchs and M. C. Whitton (2010). Improved redirection with distractors: A large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. *Virtual Reality Conference (VR), 2010 IEEE, IEEE*. doi: <https://doi.org/10.1109/VR.2010.5444816>.
- Peeters, R. R., P. Kornprobst, M. Nikolova, S. Sunaert, T. Vieville, G. Malandain, R. Deriche, O. Faugeras, M. Ng and P. Van Hecke (2004). "The use of super-resolution techniques to reduce slice thickness in functional MRI." *International Journal of Imaging Systems and Technology* **14**(3): 131-138. doi: <https://doi.org/10.1002/ima.20016>.
- Penfield, W. and E. Boldrey (1937). "Somatic motor and sensory representation in the cerebral cortex of man as studied by electrical stimulation." *Brain* **60**(4): 389-443. doi: <https://doi.org/10.1093/brain/60.4.389>.
- Penfield, W. and T. Rasmussen (1950). "The cerebral cortex of man; a clinical study of localization of function." doi: <https://doi.org/10.1001/jama.1950.02920160086033>.
- Pierce, J. S., B. C. Stearns and R. Pausch (1999). Voodoo dolls: seamless interaction at multiple scales in virtual environments. *Proceedings of the 1999 symposium on Interactive 3D graphics*. Atlanta, Georgia, USA, ACM: 141-145. doi: <https://doi.org/10.1145/300523.300540>.
- Plantard, P., E. Auvinet, A.-S. L. Pierres and F. Multon (2015). "Pose estimation with a kinect for ergonomic studies: Evaluation of the accuracy using a virtual mannequin." *Sensors* **15**(1): 1785-1803. doi: <http://dx.doi.org/10.3390/s150101785>.
- Poupyrev, I., M. Billinghamurst, S. Weghorst and T. Ichikawa (1996). The go-go interaction technique: non-linear mapping for direct manipulation in VR. *Proceedings of the 9th annual ACM symposium on User interface software and technology, ACM*. doi: <http://dx.doi.org/10.1145/237091.237102>.

- Poupyrev, I., T. Ichikawa, S. Weghorst and M. Billinghurst (1998). Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. *Computer graphics forum*, Wiley Online Library. doi: <https://doi.org/10.1111/1467-8659.00252>.
- Poupyrev, I., N. Tomokazu and S. Weghorst (1998). Virtual Notepad: handwriting in immersive VR. *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium* (Cat. No. 98CB36180), IEEE. doi: <https://doi.org/10.1109/VRAIS.1998.658467>.
- Poupyrev, I., S. Weghorst, M. Billinghurst and T. Ichikawa (1997). A framework and testbed for studying manipulation techniques for immersive VR. *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM. doi: <https://doi.org/10.1145/261135.261141>.
- Prior, A. (2006). On-the-fly voxelization for 6 degrees-of-freedom haptic virtual sculpting. *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, ACM. doi: <https://doi.org/10.1145/1128923.1128966>.
- Quek, F. K. (1996). "Unencumbered gestural interaction." *IEEE multimedia* **3**(4): 36-47. doi: <https://doi.org/10.1109/93.556459>.
- Razzaque, S., Z. Kohn and M. C. Whitton (2001). Redirected walking. *Proceedings of EUROGRAPHICS*, Manchester, UK. doi: <http://dx.doi.org/10.2312/egs.20011036>.
- Reitmayr, G. and D. Schmalstieg (2001). Mobile collaborative augmented reality. *Proceedings IEEE and ACM International Symposium on Augmented Reality*, IEEE. doi: <https://doi.org/10.1109/ISAR.2001.970521>.
- Rekimoto, J. (2002). SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *Proceedings of the SIGCHI conference*, ACM. doi: <https://doi.org/10.1145/503376.503397>.
- Ribarsky, W., J. Bolter, A. O. den Bosch and R. Van Teylingen (1994). "Visualization and analysis using virtual reality." *IEEE computer Graphics and Applications* **14**(1): 10-12. doi: <https://doi.org/10.1109/38.250911>.
- Richardson, A. E., D. R. Montello and M. Hegarty (1999). "Spatial knowledge acquisition from maps and from navigation in real and virtual environments." *Memory & cognition* **27**(4): 741-750. doi: <https://doi.org/10.3758/BF03211566>.
- Riecke, B., B. Bodenheimer, T. McNamara, B. Williams, P. Peng and D. Feuereissen (2010). "Do we need to walk for effective virtual reality navigation? physical rotations alone may suffice." *Spatial cognition VII*: 234-247. doi: [https://doi.org/10.1007/978-3-642-14749-4\\_21](https://doi.org/10.1007/978-3-642-14749-4_21).
- Rimé, B. (1982). "The elimination of visible behaviour from social interactions: Effects on verbal, nonverbal and interpersonal variables." *European journal of social psychology* **12**(2): 113-129. doi: <https://doi.org/10.1002/ejsp.2420120201>.
- Rimé, B. and L. Schiaratura (1991). "Gesture and speech."
- Robertson, R. J., F. L. Goss, J. Rutkowski, B. Lenz, C. Dixon, J. Timmer, K. Frazee, J. Dube and J. Andreacci (2003). "Concurrent validation of the OMNI perceived exertion scale for resistance exercise." *Medicine and science in sports and exercise* **35**(2): 333-341. doi: <https://doi.org/10.1249/01.MSS.0000048831.15016.2A>.



- Robles-De-La-Torre, G. and V. Hayward (2001). "Force can overcome object geometry in the perception of shape through active touch." *Nature* **412**(6845): 445-448. doi: <http://dx.doi.org/10.1038/35086588>.
- Rosa, D. A. W. and H. H. Nagel (2010). Selection techniques for dense and occluded virtual 3d environments, supported by depth feedback: Double, bound and depth bubble cursors. 2010 XXIX International Conference of the Chilean Computer Science Society, IEEE. doi: <https://doi.org/10.1109/SCCC.2010.51>.
- Sanchez-Vives, M. V. and M. Slater (2005). "From presence to consciousness through virtual reality." *Nature Reviews Neuroscience* **6**(4): 332-339. doi: <http://dx.doi.org/10.1038/nrn1651>.
- Satava, R. M. (1993). "Virtual reality surgical simulator." *Surgical endoscopy* **7**(3): 203-205. doi: <http://dx.doi.org/10.1007/BF00594110>.
- Schkolne, S., M. Pruett and P. Schröder (2001). Surface drawing: creating organic 3D shapes with the hand and tangible tools. Proceedings of the SIGCHI conference on Human factors in computing systems, ACM. doi: <https://doi.org/10.1145/365024.365114>.
- Schmalstieg, D., L. M. Encarnação and Z. Szalavári (1999). "Using transparent props for interaction with the virtual table." *SI3D* **99**: 147-153. doi: <https://doi.org/10.1145/300523.300542>.
- Seymour, N. E., A. G. Gallagher, S. A. Roman, M. K. O'Brien, V. K. Bansal, D. K. Andersen and R. M. Satava (2002). "Virtual reality training improves operating room performance: results of a randomized, double-blinded study." *Annals of surgery* **236**(4): 458-464. doi: <https://doi.org/10.1097/00000658-200210000-00008>.
- Shan, G., M. Xie, Y. Gao and X. Chi (2014). "Interactive visual exploration of halos in large-scale cosmology simulation." *Journal of Visualization* **17**(3): 145-156. doi: <https://doi.org/10.1007/s12650-014-0206-5>.
- Siciliano, B. and O. Khatib (2016). Springer handbook of robotics, Springer.
- Slater, M., M. Usoh and A. Steed (1995). "Taking steps: the influence of a walking technique on presence in virtual reality." *ACM Transactions on Computer-Human Interaction (TOCHI)* **2**(3): 201-219. doi: <https://doi.org/10.1145/210079.210084>.
- Song, D. and M. Norman (1993). Nonlinear interactive motion control techniques for virtual space navigation. Virtual Reality Annual International Symposium, 1993., 1993 IEEE, IEEE. doi: <https://doi.org/10.1109/VRAIS.1993.380790>.
- Spillmann, J., S. Tuchschnid and M. Harders (2013). "Adaptive space warping to enhance passive haptics in an arthroscopy surgical simulator." *IEEE transactions on visualization and computer graphics* **19**(4): 626-633. doi: <http://dx.doi.org/10.1109/TVCG.2013.23>.
- Sridhar, S., A. M. Feit, C. Theobalt and A. Oulasvirta (2015). Investigating the dexterity of multi-finger input for mid-air text entry. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, ACM. doi: <https://doi.org/10.1145/2702123.2702136>.
- Stanney, K. M. (2003). Virtual environments. The human-computer interaction handbook. A. J. Julie and S. Andrew, L. Erlbaum Associates Inc.: 621-634.
- Stanney, K. M., R. R. Mourant and R. S. Kennedy (1998). "Human factors issues in virtual environments: A review of the literature." *Presence* **7**(4): 327-351. doi: <https://doi.org/10.1162/105474698565767>.

- Steinicke, F., G. Bruder, J. Jerald, H. Frenz and M. Lappe (2010). "Estimation of detection thresholds for redirected walking techniques." *IEEE transactions on visualization and computer graphics* **16**(1): 17-27. doi: <https://doi.org/10.1109/TVCG.2009.62>.
- Steuer, J. (1992). "Defining virtual reality: Dimensions determining telepresence." *Journal of communication* **42**(4): 73-93. doi: <http://dx.doi.org/10.1111/j.1460-2466.1992.tb00812.x>.
- Strothoff, S., D. Valkov and K. Hinrichs (2011). Triangle cursor: Interactions with objects above the tabletop. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ACM. doi: <https://doi.org/10.1145/2076354.2076377>.
- Sturman, D. J. and D. Zeltzer (1993). "A design method for "whole-hand" human-computer interaction." *ACM Transactions on Information Systems (TOIS)* **11**(3): 219-238. doi: <https://doi.org/10.1145/159161.159159>.
- Sturman, D. J., D. Zeltzer and S. Pieper (1989). Hands-on interaction with virtual environments. *Proceedings of the 2nd annual ACM SIGGRAPH UIST*, ACM. doi: <https://doi.org/10.1145/73660.73663>.
- Sukan, M., C. Elvezio, O. Oda, S. Feiner and B. Tversky (2014). Parafrustum: Visualization techniques for guiding a user to a constrained set of viewing positions and orientations. *ACM symposium on User interface software and technology*, ACM. doi: <https://doi.org/10.1145/2642918.2647417>.
- Suma, E. A., S. Clark, D. Krum, S. Finkelstein, M. Bolas and Z. Warte (2011). Leveraging change blindness for redirection in virtual environments. *Virtual Reality Conference (VR)*, 2011 IEEE, IEEE. doi: <https://doi.org/10.1109/VR.2011.5759455>.
- Sundén, E., I. Lundgren and A. Ynnerman (2017). Hybrid Virtual Reality Touch Table: An immersive collaborative platform for public explanatory use of cultural objects and sites. *15th Eurographics Workshop on Graphics and Cultural Heritage*, Graz, Austria, September 27-29, 2017, Eurographics-European Association for Computer Graphics. doi: <https://doi.org/10.2312/gch.20171300>.
- Sutcliffe, A. (2003). *Multimedia and virtual reality: designing multisensory user interfaces*, Psychology Press.
- Sutherland, I. E. (1968). A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, ACM. doi: <https://doi.org/10.1145/1476589.1476686>.
- Szalavári, Z. and M. Gervautz (1997). The personal interaction Panel—a Two-Handed interface for augmented reality. *Computer graphics forum*, Wiley Online Library. doi: <https://doi.org/10.1111/1467-8659.00137>.
- Takahashi, K. (2017). "Pcx - Point Cloud Importer/Renderer for Unity." Available from: <https://github.com/keijiro/Pcx>.
- Tanriverdi, V. and R. J. Jacob (2001). VRID: a design model and methodology for developing virtual reality interfaces. *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM. doi: <https://doi.org/10.1145/505008.505042>.
- Templeman, J. N., P. S. Denbrook and L. E. Sibert (1999). "Virtual locomotion: Walking in place through virtual environments." *Presence: Teleoperators and virtual environments* **8**(6): 598-617. doi: <http://dx.doi.org/10.1162/105474699566512>.



- Terziman, L., M. Marchal, M. Emily, F. Multon, B. Arnaldi and A. Lécuyer (2010). Shake-your-head: Revisiting walking-in-place for desktop virtual reality. ACM Symposium on Virtual Reality Software and Technology, ACM. doi: <https://doi.org/10.1145/1889863.1889867>.
- Thalmic-Labs. <https://support.getmyo.com/hc/en-us/articles/205180865-Double-Tap-gesture-is-not-recognized>
- Thorndyke, P. W. and B. Hayes-Roth (1982). "Differences in spatial knowledge acquired from maps and navigation." *Cognitive psychology* **14**(4): 560-589. doi: [https://doi.org/10.1016/0010-0285\(82\)90019-6](https://doi.org/10.1016/0010-0285(82)90019-6).
- Tolani, D., A. Goswami and N. I. Badler (2000). "Real-time inverse kinematics techniques for anthropomorphic limbs." *Graphical models* **62**(5): 353-388. doi: <https://doi.org/10.1006/gmod.2000.0528>.
- Tregillus, S. and E. Folmer (2016). Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM. doi: <https://doi.org/10.1145/2858036.2858084>.
- Usoh, M., K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater and F. P. Brooks Jr (1999). Walking> walking-in-place> flying, in virtual environments. Conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. doi: <https://doi.org/10.1145/311535.311589>.
- Valkov, D., A. Giesler and K. H. Hinrichs (2014). Imperceptible depth shifts for touch interaction with stereoscopic objects. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM. doi: <http://dx.doi.org/10.1145/2556288.2557134>.
- Van Dam, A. (1997). "Post-WIMP user interfaces." *Communications of the ACM* **40**(2): 63-67. doi: <https://doi.org/10.1145/253671.253708>.
- Vanacken, L., T. Grossman and K. Coninx (2009). "Multimodal selection techniques for dense and occluded 3D virtual environments." *International Journal of Human-Computer Studies* **67**(3): 237-255. doi: <https://doi.org/10.1016/j.ijhcs.2008.09.001>.
- Wachs, J. P., M. Kölsch, H. Stern and Y. Edan (2011). "Vision-based hand-gesture applications." *Communications of the ACM* **54**(2): 60-71. doi: <http://dx.doi.org/10.1145/1897816.1897838>.
- Wang, J. and R. Lindeman (2014). Coordinated 3D interaction in tablet-and HMD-based hybrid virtual environments. Proceedings of the 2nd ACM symposium on Spatial user interaction, ACM. doi: <https://doi.org/10.1145/2659766.2659777>.
- Wang, J. and R. W. Lindeman (2015). Object impersonation: Towards effective interaction in tablet-and HMD-based hybrid virtual environments. Virtual Reality (VR), 2015 IEEE, IEEE. doi: <https://doi.org/10.1109/VR.2015.7223332>.
- Weissmann, J. and R. Salomon (1999). Gesture recognition for virtual reality applications using data gloves and neural networks. Neural Networks, 1999. IJCNN'99. International Joint Conference on, IEEE. doi: <https://doi.org/10.1109/IJCNN.1999.832699>.
- Wendt, J. D., M. C. Whitton and F. P. Brooks (2010). Gud wip: Gait-understanding-driven walking-in-place. Virtual Reality Conference (VR), 2010 IEEE, IEEE. doi: <https://doi.org/10.1109/VR.2010.5444812>.

- Wiktorin, C., L. Karlqvist and J. Winkel (1993). "Validity of self-reported exposures to work postures and manual materials handling." *Scandinavian journal of work, environment & health*: 208-214. <http://www.jstor.org/stable/40966137>
- Williams, B., G. Narasimham, T. P. McNamara, T. H. Carr, J. J. Rieser and B. Bodenheimer (2006). Updating orientation in large virtual environments using scaled translational gain. *Symposium on Applied Perception in Graphics and Visualization*, ACM. doi: <https://doi.org/10.1145/1140491.1140495>.
- Wilson, G., M. McGill, M. Jamieson, J. R. Williamson and S. A. Brewster (2018). Object Manipulation in Virtual Reality Under Increasing Levels of Translational Gain. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ACM. doi: <https://doi.org/10.1145/3173574.3173673>.
- Witmer, B. G. and M. J. Singer (1998). "Measuring presence in virtual environments: A presence questionnaire." *Presence* 7(3): 225-240. doi: <https://doi.org/10.1162/105474698565686>.
- Wu, M. and R. Balakrishnan (2003). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of ACM UIST*, ACM. doi: <https://doi.org/10.1145/964696.964718>.
- Xie, X., Q. Lin, H. Wu, G. Narasimham, T. P. McNamara, J. Rieser and B. Bodenheimer (2010). A system for exploring large virtual environments that combines scaled translational gain and interventions. *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, ACM. doi: <https://doi.org/10.1145/1836248.1836260>.
- Yu, L., K. Efstathiou, P. Isenberg and T. Isenberg (2012). "Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input." *IEEE transactions on visualization and computer graphics* 18(12): 2245-2254. doi: <https://doi.org/10.1109/TVCG.2012.217>.
- Yu, L., K. Efstathiou, P. Isenberg and T. Isenberg (2016). "CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds." *IEEE transactions on visualization and computer graphics* 22(1): 886-895. doi: <https://doi.org/10.1109/TVCG.2015.2467202>.
- Zank, M. and A. Kunz (2015). Using locomotion models for estimating walking targets in immersive virtual environments. *2015 International Conference on Cyberworlds (CW)*, IEEE. doi: <https://doi.org/10.1109/CW.2015.20>.
- Zhai, S. and P. Milgram (1998). Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co. doi: <http://dx.doi.org/10.1145/274644.274689>.
- Zhang, Y., J. Zhou, G. Laput and C. Harrison (2016). SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. *Proceedings of the 2016 CHI Conference*, ACM. doi: <https://doi.org/10.1145/2858036.2858082>.

## Appendix 1

### Analytical definition of Drift:

We define *drift* as the difference in the real position of the user, after traveling along a closed path in the VE. Please note *drift* is actually represented as a displacement vector.

As illustrated in **Figure 5.8**, *drift* appears as a result of the differences between the displacement performed in the VE and the real displacements of the user in reality. Let's define **A** as a period of time, in which the user has completed a closed trajectory in the VE. Thus, it can be modelled as:

$$Drift(A) = \int_{t=0}^A \left( \frac{d\mathbf{P}_W(t)}{dt} - \frac{d\mathbf{P}_T(t)}{dt} \right) \cdot d\mathbf{t} \quad (8)$$

At each point in time, the real and virtual positions of the user can be defined as:

$$\mathbf{P}_T(t) = \mathbf{M}_T^U(t) \cdot (0,0,0,1)^{\top}, \quad \forall t \in A \quad (9)$$

$$\mathbf{P}_W(t) = \mathbf{M}_W^U(t) \cdot (0,0,0,1)^{\top}, \quad \forall t \in A \quad (10)$$

Thus Eq(8) can be rewritten as:

$$Drift(A) = \int_{t=0}^A \left( \frac{d\mathbf{M}_W^U(t)}{dt} - \frac{d\mathbf{M}_T^U(t)}{dt} \right) \cdot (0,0,0,1)^{\top} \cdot d\mathbf{t} \quad (11)$$

The derivative  $d\mathbf{M}_W^U(t)$  is computed as  $d\mathbf{M}_W^U(t)/dt = (\mathbf{M}_W^U(t+dt) - \mathbf{M}_W^U(t))$ , which combined with Eq(3) allows us to rewrite it as:

$$\frac{d\mathbf{M}_W^U(t)}{dt} = D\left(k\left(\mathbf{M}_W^U(t)\right)\right) \cdot \frac{d\mathbf{M}_T^U(t)}{dt} \quad (12)$$

Combining equations (11) and (12), we can now define *drift* purely in terms of users real displacements:

$$Drift(A) = \int_{t=0}^A \left( D\left(k\left(\mathbf{M}_W^U(t)\right)\right) \cdot \frac{d\mathbf{M}_T^U(t)}{dt} - \frac{d\mathbf{M}_T^U(t)}{dt} \right) \cdot (0,0,0,1)^{\top} \cdot d\mathbf{t} \quad (13)$$

Lets use **I** to represent the identity matrix. Eq (13) can now be simplified with steps (14) and (15):

$$Drift(A) = \int_{t=0}^A \left( \mathbf{D} \left( \mathbf{k} \left( \mathbf{M}_W^U(t) \right) \right) \cdot -\mathbf{I} \right) \cdot \frac{d\mathbf{M}_T^U(t)}{dt} \cdot (0,0,0,1) \cdot d\mathbf{t} \quad (14)$$

$$Drift(A) = \int_{t=0}^A \mathbf{S}(\mathbf{k}(\mathbf{M}_W^U(t) - 1, 0), \mathbf{k}(\mathbf{M}_W^U(t) - 1)) \cdot \frac{d\mathbf{M}_T^U(t)}{dt} \cdot (0,0,0,1) \cdot d\mathbf{t} \quad (15)$$

Applying Eq(9), to represent the result in terms of users displacements, results in the final equation reported in [Chapter 5](#):

$$Drift(A) = \int_{t=0}^A \mathbf{S}(\mathbf{k}(\mathbf{M}_W^U(t) - 1, 0), \mathbf{k}(\mathbf{M}_W^U(t) - 1)) \cdot \frac{d\mathbf{P}_T(t)}{dt} \cdot d\mathbf{t} \quad (7)$$

## Appendix 2

### NaviFields: Relevance fields for adaptive VR navigation

Main effects: mean and standard deviations of the main factors in the experimental design, for all the dependent variables

#### Travelling task

	T	S	L
<b>T_D</b>	NF =0.096 ±0.2 PH =0.94±0.25	S:1=0.05±0.08 S:2=0.12±0.16 S:3=0.33±0.31	L:2=0.24±0.15 L:4=0.23±0.18
<b>T_RD</b>	NF =1.85 ±1.17 PH =0.94±0.73	S:1=0.81±0.65 S:2=1.41±0.99 S:3=1.96±1.19	L:2=0.92±0.74 L:4=1.87±1.15
<b>T_TCT</b>	NF = 3.99±3.64 PH =3.86±3.59	S:1=1.59±1.87 S:2=3.39±2.05 S:3=6.80±3.94	L:2=2.58±2.49 L:4=5.27±4.04
<b>CT</b>	NF =-1.0±1.08 PH =-0.6±1.4	S:1=-0.22±0.67 S:2=-0.67±0.94 S:3=-1.61±1.69	L:2=-0.79±1.40 L:4=-0.87±1.23
<b>ET</b>	NF = -1.0±1.20 PH =-0.5±1.61	S:1=-0.12±0.9 S:2=-0.66±1.18 S:3=-1.64±1.69	L:2=-0.7±1.47 L:4=-0.91±1.4

#### Manoeuvring task

.	T	S	H_S	T_S	P
<b>M_TCT</b>	NF=0.35 ±3.1 PH=4.53±6.09	S:1=0.07±3.13 S:2=1.63±4.12 S:3=4.99±7.8	TH =3.59±7.22 LH=1.29±3.38	TT=2.78±6.01 LT=2.1±5.45	KP=3.17±6.57 MP=2.33±5.71 SP=1.82±4.75
<b>M_OE</b>	NF=0.12 ±0.9 PH=0.4±1.07	S:1=0.15±0.93 S:2=0.30±0.96 S:3=0.44±1.13	TH =0.34±0.9 LH=0.26±1.8	TT =0.18±0.63 LT=0.41±1.28	KP=0.38±1.05 MP=0.19±1.04 SP=0.32±0.95
<b>M_PE</b>	NF=-0.8±0.8 PH =0.4±0.7	S:1=-1.02±0.7 S:2=0.69±0.7 S:3=0.44±0.8	TH =0.8±0.9 LH=-0.1±0.5	TT =-0.3±0.8 LT=-0.7±0.7	KP=0.6±0.7 MP=-1.23±0.7 SP=-0.01±0.8
<b>M_FA</b>	NF=0.25±1.94 PH=3.34±5.76	S:1=0.51±2.09 S:2=1.21±3.04 S:3=3.06±6.59	TH =2.87±5.74 LH=0.73±2.43	TT =1.7±4.25 LT=1.8±4.86	KP=2.44±5.48 MP=1.63±4.25 SP=1.33±3.72

Interaction effects: mean and standard deviation of the analysed interactions

### Travelling task

Mean = average value of the observed variable for the combination of factor indicated in the first two columns

sd = standard deviation of the observed variable for the combination of factor indicated in the first two columns

se = standard error value of the observed variable for the combination of factor indicated in the first two columns

Travelling task: T\_TCT T\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	1.74622	2.135267	0.3019724
_2	NAVIFIELD	1.44690	1.576119	0.2228969
_4	HOMOGENEOUS	3.11976	2.588654	0.3660909
_4	NAVIFIELD	3.66088	2.421883	0.3425059
_8	HOMOGENEOUS	6.72608	3.806978	0.5383880
_8	NAVIFIELD	6.88304	4.110387	0.5812966

Travelling task: T\_TCT T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	2.488560	2.484394	0.2868731
2	NAVIFIELD	2.691427	2.516910	0.2906278
4	HOMOGENEOUS	5.239480	3.998171	0.4616690
4	NAVIFIELD	5.302453	4.126104	0.4764415

Travelling task: T\_D T\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	0.07900	0.09395809	0.01328768
_2	NAVIFIELD	0.02422	0.07744235	0.01095200
_4	HOMOGENEOUS	0.19740	0.17871046	0.02527348
_4	NAVIFIELD	0.06042	0.10341537	0.01462514
_8	HOMOGENEOUS	0.46792	0.26472487	0.03743775
_8	NAVIFIELD	0.20566	0.30009424	0.04243974

Travelling task: T\_D T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	0.23045333	0.2681925	0.03096820
2	NAVIFIELD	0.08802667	0.1954737	0.02257136
4	HOMOGENEOUS	0.26576000	0.2335490	0.02696792
4	NAVIFIELD	0.10550667	0.2116147	0.02443516

Travelling task: T\_RD T\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	0.76434	0.7034227	0.09947899
_2	NAVIFIELD	0.86772	0.6025367	0.08521156
_4	HOMOGENEOUS	0.83928	0.6689246	0.09460023
_4	NAVIFIELD	1.99196	0.9300857	0.13153398
_8	HOMOGENEOUS	1.21934	0.7463257	0.10554639
_8	NAVIFIELD	2.71572	1.0839224	0.15328978

Travelling task: T\_rd T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	0.5978667	0.5417878	0.06256027
2	NAVIFIELD	1.2543600	0.7714857	0.08908350
4	HOMOGENEOUS	1.2841067	0.7355293	0.08493160
4	NAVIFIELD	2.4625733	1.1950077	0.13798761

Travelling task: T\_CT T\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	-0.30	0.7889544	0.11157500
_2	NAVIFIELD	-0.14	0.5349041	0.07564687
_4	HOMOGENEOUS	-0.88	1.0427593	0.14746843
_4	NAVIFIELD	-0.46	0.7879190	0.11142857
_8	HOMOGENEOUS	-1.92	1.9149253	0.27081133
_8	NAVIFIELD	-1.30	1.4033488	0.19846349

Travelling task: T\_CT T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	-1.0000000	1.576860	0.1820801
2	NAVIFIELD	-0.5866667	1.175117	0.1356909
4	HOMOGENEOUS	-1.0666667	1.407829	0.1625621
4	NAVIFIELD	-0.6800000	1.002160	0.1157194

Travelling task: T\_ET T\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	-0.24	0.9595917	0.1357068
_2	NAVIFIELD	0.00	0.8329931	0.1178030
_4	HOMOGENEOUS	-0.98	1.3013337	0.1840364
_4	NAVIFIELD	-0.34	0.9606545	0.1358571
_8	HOMOGENEOUS	-1.96	1.9373873	0.2739879
_8	NAVIFIELD	-1.32	1.3618715	0.1925977

Travelling task: T\_ET T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	-0.9466667	1.609879	0.1858929
2	NAVIFIELD	-0.4533333	1.287081	0.1486193
4	HOMOGENEOUS	-1.1733333	1.613904	0.1863575
4	NAVIFIELD	-0.6533333	1.120971	0.1294386

**Manoeuvring task**

Head: 1 = Large, 2=Tight

Tail: 1 = Large, 2=Tight

Position: 1=Kneeling, 2=Medium, 3=Standing

Manoeuvring task: m\_tct\*s

scaling	technique	mean	sd	se
_2	HOMOGENEOUS	1.0629826	2.896446	0.1706747
_2	NAVIFIELD	0.3559688	3.324970	0.1959257
_4	HOMOGENEOUS	2.8306771	4.434044	0.2612786
_4	NAVIFIELD	0.4455556	3.406197	0.2007121
_8	HOMOGENEOUS	9.7172639	8.564749	0.5046827
_8	NAVIFIELD	0.2639861	2.604628	0.1534792

Manoeuvring task: m\_tct\*H\_S

head	technique	mean	sd	se
1	HOMOGENEOUS	2.3832986	4.112320	0.19785407
1	NAVIFIELD	0.2155856	1.920408	0.09239566
2	HOMOGENEOUS	6.6906505	8.322096	0.40039703
2	NAVIFIELD	0.4947546	3.986103	0.19178147

Manoeuvring task: m\_tct\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	4.0610556	6.694000	0.3220652
1	NAVIFIELD	0.1584583	2.691600	0.1294997
2	HOMOGENEOUS	5.0128935	7.085402	0.3408966
2	NAVIFIELD	0.5518819	3.506246	0.1686943

Manoeuvring task: m\_tct\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	5.7454514	7.829875	0.4613798
1	NAVIFIELD	0.6063368	3.475145	0.2047749
2	HOMOGENEOUS	4.5213576	6.872677	0.4049764
2	NAVIFIELD	0.1550243	2.926251	0.1724310
3	HOMOGENEOUS	3.3441146	5.650839	0.3329789
3	NAVIFIELD	0.3041493	2.951702	0.1739307

Manoeuvring task: m\_oe\*s

scaling	technique	mean	sd	se
_2	HOMOGENEOUS	0.2057986	0.9740630	0.05739721
_2	NAVIFIELD	0.1131458	0.8938778	0.05267226
_4	HOMOGENEOUS	0.4858854	0.9665850	0.05695657
_4	NAVIFIELD	0.1339757	0.9368965	0.05520715
_8	HOMOGENEOUS	0.7422951	1.2044388	0.07097224
_8	NAVIFIELD	0.1415069	0.9842651	0.05799838



Manoeuvring task: m\_oe\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	0.5592685	0.9728240	0.04680501
1	NAVIFIELD	0.1353565	0.8916923	0.04290157
2	HOMOGENEOUS	0.3967176	1.1647002	0.05603666
2	NAVIFIELD	0.1237292	0.9832461	0.04730645

Manoeuvring task: m\_oe\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	0.60945370	1.3640550	0.06562813
1	NAVIFIELD	0.22952546	1.1804258	0.05679326
2	HOMOGENEOUS	0.34653241	0.6487409	0.03121256
2	NAVIFIELD	0.02956019	0.5903486	0.02840316

Manoeuvring task: m\_oe\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	0.55415625	1.1008232	0.06486663
1	NAVIFIELD	0.21067014	0.9804536	0.05777378
2	HOMOGENEOUS	0.37702083	1.0920271	0.06434831
2	NAVIFIELD	0.02086458	0.9701661	0.05716758
3	HOMOGENEOUS	0.50280208	1.0282330	0.06058921
3	NAVIFIELD	0.15709375	0.8510427	0.05014817

Manoeuvring task: m\_pe\*s

scaling	technique	mean	sd	se
_2	HOMOGENEOUS	-0.0009826389	0.007630151	0.0004496110
_2	NAVIFIELD	-0.0010763889	0.008086708	0.0004765138
_4	HOMOGENEOUS	0.0007534722	0.007860167	0.0004631648
_4	NAVIFIELD	-0.0007534722	0.007907897	0.0004659773
_8	HOMOGENEOUS	0.0016631944	0.007131011	0.0004201989
_8	NAVIFIELD	-0.0007812500	0.008680391	0.0005114970

Manoeuvring task: m\_pe\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	-0.0002638889	0.009665870	0.0004650494
1	NAVIFIELD	-0.0021597222	0.010123137	0.0004870497
2	HOMOGENEOUS	0.0012199074	0.004653195	0.0002238770
2	NAVIFIELD	0.0004189815	0.005440069	0.0002617354

Manoeuvring task: m\_pe\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	0.0003495370	0.007604221	0.0003658583
1	NAVIFIELD	-0.0004976852	0.007806535	0.0003755921
2	HOMOGENEOUS	0.0006064815	0.007637237	0.0003674467
2	NAVIFIELD	-0.0012430556	0.008613037	0.0004143949

Manoeuvring task: m\_pe\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	0.0007847222	0.007126691	0.0004199443
1	NAVIFIELD	0.0005694444	0.007812393	0.0004603497
2	HOMOGENEOUS	-0.0004409722	0.007253738	0.0004274306
2	NAVIFIELD	-0.0018020833	0.007976822	0.0004700387
3	HOMOGENEOUS	0.0010902778	0.008351939	0.0004921427
3	NAVIFIELD	-0.0013784722	0.008687318	0.0005119051

Manoeuvring task: m\_fa\*s

scaling	technique	mean	sd	se
_2	HOMOGENEOUS	0.7256944	1.980154	0.11668169
_2	NAVIFIELD	0.2986111	2.181902	0.12856979
_4	HOMOGENEOUS	2.1111111	3.511278	0.20690405
_4	NAVIFIELD	0.3194444	2.156435	0.12706916
_8	HOMOGENEOUS	7.2118056	7.755874	0.45701928
_8	NAVIFIELD	0.1493056	1.407521	0.08293895

Manoeuvring task: m\_fa\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	1.40509259	3.062424	0.1473410
1	NAVIFIELD	0.05787037	1.204740	0.0579631
2	HOMOGENEOUS	5.29398148	7.036475	0.3385426
2	NAVIFIELD	0.45370370	2.463221	0.1185118

Manoeuvring task: m\_fa\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	3.1643519	5.434984	0.26149079
1	NAVIFIELD	0.2847222	1.587348	0.07637131
2	HOMOGENEOUS	3.5347222	6.071400	0.29211038
2	NAVIFIELD	0.2268519	2.253004	0.10839773

Manoeuvring task: m\_fa\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	4.4375000	6.806657	0.40108609
1	NAVIFIELD	0.4444444	2.447434	0.14421644
2	HOMOGENEOUS	3.1111111	5.384985	0.31731329
2	NAVIFIELD	0.1493056	1.695022	0.09988015
3	HOMOGENEOUS	2.5000000	4.745803	0.27964914
3	NAVIFIELD	0.1736111	1.578734	0.09302782

Manoeuvring task: m\_CM\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	-0.52	1.249326	0.1766814
_2	NAVIFIELD	-0.52	1.297407	0.1834811
_4	HOMOGENEOUS	-2.38	1.828237	0.2585517
_4	NAVIFIELD	-0.52	1.164790	0.1647261
_8	HOMOGENEOUS	-4.08	1.688376	0.2387724
_8	NAVIFIELD	-0.56	1.214874	0.1718091

Maneuvering task: m\_CM\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	-2.6133333	2.098219	0.2422814
2	NAVIFIELD	-0.6933333	1.173737	0.1355314
4	HOMOGENEOUS	-2.0400000	2.202210	0.2542893
4	NAVIFIELD	-0.3733333	1.249577	0.1442887

Manoeuvring task: m\_EM\*S

scale	technique	mean	sd	se
_2	HOMOGENEOUS	-0.50	1.249490	0.1767045
_2	NAVIFIELD	-0.26	1.157231	0.1636572
_4	HOMOGENEOUS	-2.08	1.601530	0.2264905
_4	NAVIFIELD	-0.46	1.128662	0.1596169
_8	HOMOGENEOUS	-3.38	1.904774	0.2693757
_8	NAVIFIELD	-0.60	1.124858	0.1590790

Manoeuvring task: m\_EM\*T

length	technique	mean	sd	se
2	HOMOGENEOUS	-2.1466667	2.128909	0.2458252
2	NAVIFIELD	-0.4666667	1.154701	0.1333333
4	HOMOGENEOUS	-1.8266667	1.833423	0.2117055
4	NAVIFIELD	-0.4133333	1.128181	0.1302711

**Baseline Raw data:****Travelling task**

Travelling task: T\_TCT T\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	7.28948	3.273871	0.4629952

Travelling task: T\_TCT T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	4.73496	1.289355	0.2578709
4	HOMOGENEOUS	9.84400	2.574038	0.5148076

Travelling task: T\_D T\*S

scale	technique	mean	sd	se
M NONE	HOMOGENEOUS	0.12352	0.04278052	0.006050079

Travelling task: T\_D T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	0.12824	0.04957257	0.009914515
4	HOMOGENEOUS	0.11880	0.03511054	0.007022108

Travelling task: T\_RD T\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	3.99112	1.444545	0.2042896

Travelling task: T\_rd T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	2.64784	0.3073147	0.06146293
4	HOMOGENEOUS	5.33440	0.6377752	0.12755505

Travelling task: T\_CT T\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	5.68	0.6207303	0.08778452

Travelling task: T\_CT T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	5.60	0.6454972	0.1290994
4	HOMOGENEOUS	5.76	0.5972158	0.1194432

Travelling task: T\_ET T\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	5.4	0.9035079	0.1277753

Travelling task: T\_ET T\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	5.36	0.9073772	0.1814754
4	HOMOGENEOUS	5.44	0.9165151	0.1833030

**Manoeuvring task**

Manoeuvring task: m\_tct\*s

scaling	technique	mean	sd	se
M_NONE	HOMOGENEOUS	4.969264	1.999873	0.1178436

Manoeuvring task: m\_tct\*H\_S

head	technique	mean	sd	se
1	HOMOGENEOUS	4.321271	1.247100	0.1039250
2	HOMOGENEOUS	5.617257	2.371926	0.1976605

Manoeuvring task: m\_tct\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	5.058931	1.961935	0.1634946
2	HOMOGENEOUS	4.879597	2.039993	0.1699994

Manoeuvring task: m\_tct\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	5.315760	2.101123	0.2144450
2	HOMOGENEOUS	4.982490	1.810279	0.1847609
3	HOMOGENEOUS	4.609542	2.034351	0.2076301

Manoeuvring task: m\_oe\*s

scaling	technique	mean	sd	se
M_NONE	HOMOGENEOUS	1.683458	1.004471	0.059189

Manoeuvring task: m\_oe\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	1.702583	0.9055118	0.07545931
2	HOMOGENEOUS	1.664333	1.0974015	0.09145012

Manoeuvring task: m\_oe\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	1.996938	1.2001136	0.10000947
2	HOMOGENEOUS	1.369979	0.6219267	0.05182722

Manoeuvring task: m\_oe\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	1.651260	0.7415302	0.07568211
2	HOMOGENEOUS	1.719510	1.1387441	0.11622258
3	HOMOGENEOUS	1.679604	1.0950543	0.11176351

Manoeuvring task: m\_pe\*s

scaling	technique	mean	sd	se
M_NONE	HOMOGENEOUS	0.02332986	0.009915262	0.0005842624

Manoeuvring task: m\_pe\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	0.03078472	0.008248893	0.0006874078
2	HOMOGENEOUS	0.01587500	0.004164242	0.0003470201

Manoeuvring task: m\_pe\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	0.02336806	0.009861096	0.0008217580
2	HOMOGENEOUS	0.02329167	0.010003409	0.0008336174

Manoeuvring task: m\_pe\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	0.02343750	0.009484405	0.0009679981
2	HOMOGENEOUS	0.02376042	0.010156403	0.0010365835
3	HOMOGENEOUS	0.02279167	0.010169010	0.0010378703

Manoeuvring task: m\_fa\*s

scaling	technique	mean	sd	se
M_NONE	HOMOGENEOUS	1.361111	1.163385	0.06855313

Manoeuvring task: m\_fa\*H\_s

head	technique	mean	sd	se
1	HOMOGENEOUS	1.048611	0.7784678	0.06487231
2	HOMOGENEOUS	1.673611	1.3833670	0.11528059

Manoeuvring task: m\_fa\*T\_s

tail	technique	mean	sd	se
1	HOMOGENEOUS	1.354167	1.060866	0.08840552
2	HOMOGENEOUS	1.368056	1.261293	0.10510777

Manoeuvring task: m\_fa\*P

position	technique	mean	sd	se
1	HOMOGENEOUS	1.281250	1.033167	0.1054472
2	HOMOGENEOUS	1.354167	1.151467	0.1175212
3	HOMOGENEOUS	1.447917	1.296715	0.1323454

Manoeuvring task: m\_CM\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	5.28	1.178723	0.1666966

Manoeuvring task: m\_CM\*L

length	technique	mean	sd	se
2	HOMOGENEOUS	5.52	0.7141428	0.1428286
4	HOMOGENEOUS	5.04	1.4854853	0.2970971

Manoeuvring task: m\_EM\*S

scale	technique	mean	sd	se
M_NONE	HOMOGENEOUS	4.98	1.220405	0.1725913

Manoeuvring task: m\_EM\*T

length	technique	mean	sd	se
2	HOMOGENEOUS	5.12	1.092398	0.2184796
4	HOMOGENEOUS	4.84	1.344123	0.2688246

Comparison between NaviFields and baseline: Bonferroni corrected pairwise comparisons

### Travelling task

Baseline is expressed as absence of scaling (\_NONE). Values in the tables represents p-values for Bonferroni corrected paired t-test, computed using the R software for statistical computing (pairwise.t.test). Combination of columns and rows indicate the compared scalings.

Travelling task: T\_tct

	_2	_4	_8
_4	1.4e-08	-	-
_8	4.7e-13	1.9e-07	-
_NONE	2.4e-07	1.3e-13	3.3e-15

Travelling task: T\_d

	_2	_4	_8
_4	0.23171	-	-
_8	0.00070	0.00214	-
_NONE	0.19019	0.00084	7.9e-05

Travelling task: T\_rd

	_2	_4	_8
_4	9.1e-16	-	-
_8	< 2e-16	5.1e-09	-
_NONE	6.6e-13	< 2e-16	< 2e-16

Travelling task: T\_ct

	_2	_4	_8
_4	0.02938	-	-
_8	6.0e-07	4.4e-05	-
_NONE	0.42147	0.00085	2.0e-07

Travelling task: T\_et

	_2	_4	_8
_4	0.070	-	-
_8	1.8e-09	3.1e-07	-
_NONE	1.000	0.094	6.7e-08

**Manoeuvring task**

Manoeuvring task: M\_tct

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	1.00	-	-
<u>8</u>	1.00	1.00	-
<u>NONE</u>	0.42	0.16	0.52

Manoeuvring task: M\_oe

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	1.000	-	-
<u>8</u>	1.000	1.000	-
<u>NONE</u>	0.195	0.095	0.092

Manoeuvring task: M\_pe

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	1.00	-	-
<u>8</u>	1.00	1.00	-
<u>NONE</u>	0.15	0.64	0.77

Manoeuvring task: M\_FA

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	1.00	-	-
<u>8</u>	1.00	1.00	-
<u>NONE</u>	0.22	0.16	1.00

Manoeuvring task: M\_cm

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	1.00	-	-
<u>8</u>	1.00	1.00	-
<u>NONE</u>	0.22	0.22	0.18

Manoeuvring task: M\_em

	<u>2</u>	<u>4</u>	<u>8</u>
<u>4</u>	0.95	-	-
<u>8</u>	0.76	0.95	-
<u>NONE</u>	0.95	0.38	0.13



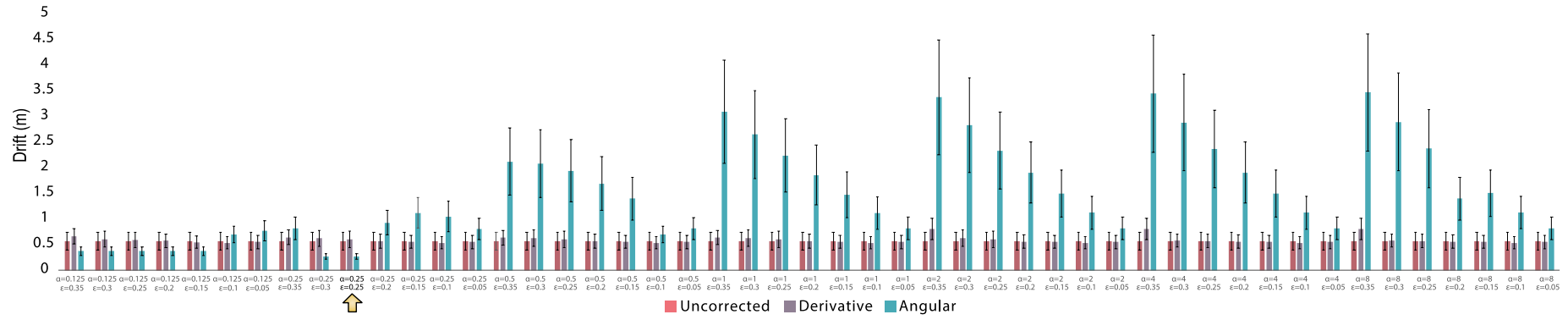
## Appendix 3

We conducted paired comparisons with Bonferroni corrections to compare the *Drift* effect with both correction techniques (*Derivative* and *Angular*) in each configuration of  $\alpha$  and  $\varepsilon$  (49 configurations in total). **Table S1** shows the mean and  $p$  values of this comparison. Results suggest a poor performance of the *Derivative* correction when compared with the uncorrected *Drift*. Averages with SD of all the configuration are shown in **Figure S1**.

	$\alpha$	$\varepsilon$	Uncorrected	Derivative	Angular	Uncorrected Vs Derivative	Uncorrected Vs Angular	Derivative Vs Angular
1	0.125	0.35	0.566 m	0.665 m	0.374 m	$p=0.037$	$p<0.001$	$p<0.001$
2	0.125	0.3	0.566 m	0.608 m	0.374 m	$p=0.891$	$p<0.001$	$p<0.001$
3	0.125	0.25	0.566 m	0.591 m	0.374 m	$p=1$	$p<0.001$	$p<0.001$
4	0.125	0.2	0.566 m	0.573 m	0.374 m	$p=1$	$p<0.001$	$p=0.002$
5	0.125	0.15	0.566 m	0.546 m	0.374 m	$p=1$	$p<0.001$	$p=0.003$
6	0.125	0.1	0.566 m	0.533 m	0.697 m	$p=0.339$	$p<0.001$	$p<0.001$
7	0.125	0.05	0.566 m	0.550 m	0.770 m	$p=0.906$	$p<0.001$	$p<0.001$
8	0.25	0.35	0.566 m	0.644 m	0.816 m	$p=0.253$	$p<0.001$	$p<0.001$
9	0.25	0.3	0.566 m	0.623 m	0.268 m	$p=0.496$	$p<0.001$	$p<0.001$
10	0.25	0.25	0.566 m	0.603 m	0.268 m	$p=0.765$	$p<0.001$	$p<0.001$
11	0.25	0.2	0.566 m	0.564 m	0.929 m	$p=1$	$p<0.001$	$p<0.001$
12	0.25	0.15	0.566 m	0.554 m	1.118 m	$p=1$	$p<0.001$	$p<0.001$
13	0.25	0.1	0.566 m	0.536 m	1.047 m	$p=0.282$	$p<0.001$	$p<0.001$
14	0.25	0.05	0.566 m	0.550 m	0.805 m	$p=0.908$	$p<0.001$	$p<0.001$
15	0.5	0.35	0.566 m	0.634 m	2.118 m	$p=0.432$	$p<0.001$	$p<0.001$
16	0.5	0.3	0.566 m	0.627 m	2.082 m	$p=0.306$	$p<0.001$	$p<0.001$
17	0.5	0.25	0.566 m	0.606 m	1.942 m	$p=0.53$	$p<0.001$	$p<0.001$
18	0.5	0.2	0.566 m	0.569 m	1.687 m	$p=1$	$p<0.001$	$p<0.001$
19	0.5	0.15	0.566 m	0.555 m	1.396 m	$p=1$	$p<0.001$	$p<0.001$
20	0.5	0.1	0.566 m	0.536 m	0.697 m	$p=0.289$	$p<0.001$	$p<0.001$
21	0.5	0.05	0.566 m	0.550 m	0.814 m	$p=0.907$	$p<0.001$	$p<0.001$
22	1	0.35	0.566 m	0.635 m	3.088 m	$p=0.385$	$p<0.001$	$p<0.001$
23	1	0.3	0.566 m	0.629 m	2.648 m	$p=0.311$	$p<0.001$	$p<0.001$
24	1	0.25	0.566 m	0.606 m	2.239 m	$p=0.503$	$p<0.001$	$p<0.001$
25	1	0.2	0.566 m	0.570 m	1.854 m	$p=1$	$p<0.001$	$p<0.001$
26	1	0.15	0.566 m	0.552 m	1.475 m	$p=1$	$p<0.001$	$p<0.001$
27	1	0.1	0.566 m	0.536 m	1.117 m	$p=0.285$	$p<0.001$	$p<0.001$
28	1	0.05	0.566 m	0.550 m	0.815 m	$p=0.908$	$p<0.001$	$p<0.001$
29	2	0.35	0.566 m	0.807 m	3.371 m	$p<0.001$	$p<0.001$	$p<0.001$
30	2	0.3	0.566 m	0.628 m	2.830 m	$p=0.344$	$p<0.001$	$p<0.001$
31	2	0.25	0.566 m	0.606 m	2.332 m	$p=0.503$	$p<0.001$	$p<0.001$
32	2	0.2	0.566 m	0.559 m	1.902 m	$p=1$	$p<0.001$	$p<0.001$
33	2	0.15	0.566 m	0.552 m	1.494 m	$p=1$	$p<0.001$	$p<0.001$
34	2	0.1	0.566 m	0.536 m	1.123 m	$p=0.284$	$p<0.001$	$p<0.001$
35	2	0.05	0.566 m	0.550 m	0.816 m	$p=0.909$	$p<0.001$	$p<0.001$
36	4	0.35	0.566 m	0.806 m	3.445 m	$p<0.001$	$p<0.001$	$p<0.001$
37	4	0.3	0.566 m	0.582 m	2.882 m	$p=1$	$p<0.001$	$p<0.001$
38	4	0.25	0.566 m	0.572 m	2.363 m	$p=1$	$p<0.001$	$p<0.001$
39	4	0.2	0.566 m	0.558 m	1.904 m	$p=1$	$p<0.001$	$p<0.001$
40	4	0.15	0.566 m	0.552 m	1.500 m	$p=1$	$p<0.001$	$p<0.001$
41	4	0.1	0.566 m	0.536 m	1.125 m	$p=0.286$	$p<0.001$	$p<0.001$
42	4	0.05	0.566 m	0.550 m	0.816 m	$p=0.909$	$p<0.001$	$p<0.001$
43	8	0.35	0.566 m	0.807 m	3.468 m	$p<0.001$	$p<0.001$	$p<0.001$
44	8	0.3	0.566 m	0.582 m	2.895 m	$p=1$	$p<0.001$	$p<0.001$
45	8	0.25	0.566 m	0.572 m	2.374 m	$p=1$	$p<0.001$	$p<0.001$
46	8	0.2	0.566 m	0.559 m	1.396 m	$p=1$	$p<0.001$	$p<0.001$
47	8	0.15	0.566 m	0.552 m	1.502 m	$p=1$	$p<0.001$	$p<0.001$
48	8	0.1	0.566 m	0.536 m	1.126 m	$p=0.286$	$p<0.001$	$p<0.001$
49	8	0.05	0.566 m	0.550 m	0.816 m	$p=0.886$	$p<0.001$	$p<0.001$

**Table S1.** Mean *Drift* of each technique (Uncorrected, *Derivative* and *Angular*) in meters and results from our paired comparisons with Bonferroni corrections to compare the *Drift* effect in each of the 49 configurations of  $\alpha$  and  $\varepsilon$ . The highlighted row shows the best configuration.

**Figure S1** depicts the average of uncorrected *Drift* compared with the two corrections techniques (*Angular* and *Derivative*) in each of the 49 configurations of  $\alpha$  and  $\epsilon$ . The lower *Drift* was resultant from the configuration  $\epsilon=0.25$  and  $\alpha=0.25$  which was significantly different from both uncorrected and *Derivative* correction (see Table S1 for paired comparisons).



**Figure S1.** Mean *Drift* uncorrected (red bars) and resultant *Drift* with *Derivative* (purple bars) and *Angular* (green bars) correction. Error bars represents SD. The yellow arrow highlights the best configuration ( $\epsilon=0.25$  and  $\alpha=0.25$ ) with lower *Drift*.

## Generability And Robustness of Our Correction Technique

### Drift Correction

We measured the average Drift resulting from our *Angular* correction in each navigation technique (*NaviFields* and *Seven League Boots*). **Table S2** shows the average *Drift* and SD in meters along with a comparison of the original technique (*NF* and *7LB*) against their corrected counterpart (*NFC* and *7LBC*). Due to participants were not able to reach the last sets of flags in the uncorrected technique, we added a label indicating not enough data to conduct the analysis.

Flag	NaviFields				7 League Boots			
	NF	NFC	NF Vs NFC	Effect Size	7LB	7LBC	7LB Vs 7LBC	Effect Size
7	<b><i>M=0.621</i></b> <i>SD=0.21</i>	<b><i>M=0.259</i></b> <i>SD=0.13</i>	<i>p=0.003</i>	$\eta^2=0.604$	<b><i>M=0.508</i></b> <i>SD=0.25</i>	<b><i>M=0.255</i></b> <i>SD=0.06</i>	<i>p=0.006</i>	$\eta^2=0.587$
14	<b><i>M=0.743</i></b> <i>SD=0.23</i>	<b><i>M=0.588</i></b> <i>SD=0.12</i>	<i>p=0.089</i>	$\eta^2=0.288$	<b><i>M=0.527</i></b> <i>SD=0.24</i>	<b><i>M=0.198</i></b> <i>SD=0.02</i>	<i>p=0.012</i>	$\eta^2=0.679$
21	<b><i>M=0.856</i></b> <i>SD=0.45</i>	<b><i>M=0.236</i></b> <i>SD=0.04</i>	<i>p=0.33</i>	$\eta^2=0.755$	<b><i>M=0.561</i></b> <i>SD=0.25</i>	<b><i>M=0.236</i></b> <i>SD=0.02</i>	<i>p=0.295</i>	$\eta^2=0.800$
28	<b><i>M=1.32</i></b> Not enough data	<b><i>M=0.172</i></b> <i>SD=0.11</i>	Not enough data	Not enough data	<b><i>M=0.433</i></b> Not enough data	<b><i>M=0.208</i></b> <i>SD=0.106</i>	Not enough data	Not enough data
35	Not enough data	<b><i>M=0.167</i></b> <i>SD=0.08</i>	Not enough data	Not enough data	<b><i>M=0.384</i></b> Not enough data	<b><i>M=0.188</i></b> <i>SD=0.05</i>	Not enough data	Not enough data
42	Not enough data	<b><i>M=0.195</i></b> <i>SD=0.08</i>	Not enough data	Not enough data	<b><i>M=0.183</i></b> Not enough data	<b><i>M=0.167</i></b> <i>SD=0.04</i>	Not enough data	Not enough data

**Table S2. Summary of *Drift* effects in each correction technique and comparison uncorrected vs corrected.**

### Performance

We also evaluated participants performance in terms of Flags Reached (FR), Real Distance per Flag (RDF) and Time per Flag (TPF). **Table S3** shows the results from a One-way Repeated Measures ANOVA analysis to compare the effect of the correction techniques on participants performance in both original navigation methods (*NF* and *SLB*) and their corrected counterpart (*NFC* and *7LBC*).

Performance	NaviFields				7 League Boots			
	NF	NFC	NF Vs NFC	Effect Size	7LB	7LBC	7LB Vs 7LBC	Effect Size
FR	<b><i>M=9.13</i></b> <i>SD=4.31</i>	<b><i>M=41.02</i></b> <i>SD=2.17</i>	<i>p&lt;0.001</i>	$\eta^2=0.983$	<b><i>M=10.52</i></b> <i>SD=5.097</i>	<b><i>M=36.61</i></b> <i>SD=9.44</i>	<i>p&lt;0.001</i>	$\eta^2=0.893$
RDF	<b><i>M=2.931</i></b> <i>SD=1.09</i>	<b><i>M=3.386</i></b> <i>SD=0.23</i>	<i>p=0.215</i>	$\eta^2=0.136$	<b><i>M=4.16</i></b> <i>SD=2.32</i>	<b><i>M=3.65</i></b> <i>SD=0.85</i>	<i>p=0.492</i>	$\eta^2=0.044$
TPF	<b><i>M=3.653</i></b> <i>SD=1.22</i>	<b><i>M=3.510</i></b> <i>SD=0.91</i>	<i>p=0.692</i>	$\eta^2=0.015$	<b><i>M=3.90</i></b> <i>SD=2.17</i>	<b><i>M=3.463</i></b> <i>SD=0.70</i>	<i>p=0.521</i>	$\eta^2=0.038$

**Table S3. Summary of effects of the correction techniques on performance.**

### User Experience

We conducted a subjective evaluation of participants experience. Participants were asked to rate the level of agreement regarding 8 statements (shown in **Table S4**) in a Likert scale form 1 (totally disagree) to 7 (totally agree). A One-way Repeated Measures ANOVA was conducted to compare the effect of the *Angular* correction on participants' subjective experience in both navigation methods (*NF* and *7LB*). **Table S5** shows mean, SD and effects in each statement between uncorrected (*NF* and *7LB*) and corrected techniques (*NFC* and *7LBC*).

1	Moving from flag to flag was comfortable.
2	Moving from flag to flag was easy.
3	I felt control of my movements in the virtual world.
4	I felt involved in the VR experience.
5	The interaction with the VE seems natural.
6	The experiences in the virtual world seems consistent with the real-world.
7	The sense of moving around inside the VE was compelling.
8	I felt very proficient in moving and interacting with the VE.

**Table S4. List of statements used in our subjective evaluation.**

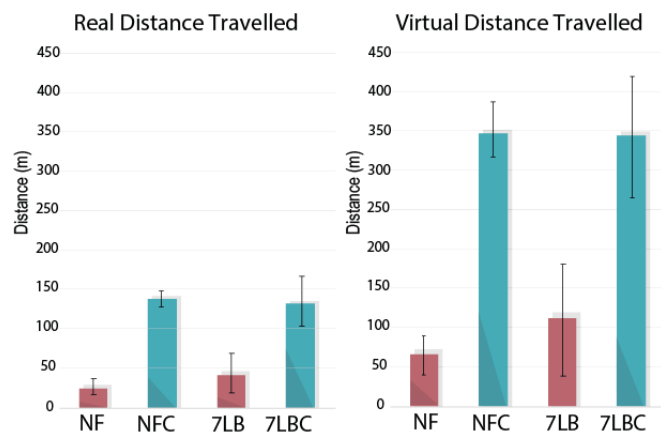
Question	NaviFields				7 league boots			
	NF	NFC	NF Vs NFC	Effect Size	7LB	7LBC	7LB Vs 7LBC	Effect Size
Comfortable	<b>M=3.58</b> SD=1.56	<b>M=3.67</b> SD=1.82	$p=0.77$	$\eta^2=0.008$	<b>M=3.92</b> SD=1.56	<b>M=5.17</b> SD=0.93	$p=0.002$	$\eta^2=0.605$
Easy	<b>M=3.5</b> SD=1.24	<b>M=4.08</b> SD=1.62	$p=0.089$	$\eta^2=0.240$	<b>M=4.00</b> SD=1.27	<b>M=5.33</b> SD=0.49	$p=0.006$	$\eta^2=0.508$
Control	<b>M=3.42</b> SD=1.50	<b>M=3.92</b> SD=1.83	$p=0.339$	$\eta^2=0.083$	<b>M=3.75</b> SD=1.81	<b>M=5.25</b> SD=0.75	$p=0.023$	$\eta^2=0.386$
Involved	<b>M=4.17</b> SD=1.52	<b>M=4.33</b> SD=1.55	$p=0.551$	$\eta^2=0.033$	<b>M=4.5</b> SD=1.38	<b>M=5.08</b> SD=0.9	$p=0.067$	$\eta^2=0.272$
Natural	<b>M=2.67</b> SD=1.77	<b>M=3.33</b> SD=1.87	$p=0.180$	$\eta^2=0.157$	<b>M=3.67</b> SD=1.61	<b>M=4.92</b> SD=0.99	$p=0.011$	$\eta^2=0.457$
Consistent	<b>M=2.58</b> SD=1.67	<b>M=2.92</b> SD=1.83	$p=0.305$	$\eta^2=0.095$	<b>M=3.08</b> SD=2.06	<b>M=4.33</b> SD=1.15	$p=0.028$	$\eta^2=0.368$
Sense of Moving	<b>M=3.67</b> SD=1.37	<b>M=4.00</b> SD=1.95	$p=0.22$	$\eta^2=0.133$	<b>M=4.5</b> SD=1.508	<b>M=5.17</b> SD=0.71	$p=0.087$	$\eta^2=0.242$
Proficient	<b>M=3.67</b> SD=1.303	<b>M=4.17</b> SD=0.93	$p=0.053$	$\eta^2=0.300$	<b>M=3.83</b> SD=1.8	<b>M=5.17</b> SD=0.83	$p=0.018$	$\eta^2=0.410$

**Table S5. Summary of effects of the correction techniques on participants user experience.**

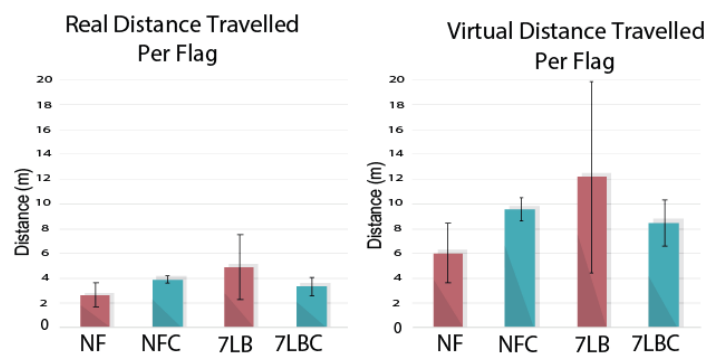
### Travelled Distances

ID	REAL DISTANCE				VIRTUAL DISTANCE			
	NF	NFC	7LB	7LBC	NF	NFC	7LB	7LBC
1	37.66	125.71	10.202	164.338	100.454	323.8	19.2446	353.152
2	21.8773	128.572	49.932	153.505	54.9746	319.56	114.686	352.457
3	40.3116	135.873	63.626	142.065	97.4556	323.875	147.503	335.862
4	18.0816	144.462	22.491	66.3553	42.506	333.618	40.9063	123.584
5	38.068	151.480	24.288	156.094	94.74	351.340	50.0786	377.894
6	26.204	132.012	99.151	139.989	58.1476	297.512	236.357	363.430
7	17.5846	146.306	65.516	137.536	47.4203	402.262	196.189	412.710
8	16.618	134.093	21.887	153.311	47.4203	402.262	196.189	412.710
9	34.37	126.849	50.930	74.3376	43.2996	346.22	52.966	404.983
10	35.78366	139.495	43.634	142.590	83.648	331.476	102.972	334.773
11	23.178	149.845	41.427	125.875	55.9196	378.076	105.552	322.242
12	11.775	152.541	32.9906	147.6173	28.5016	365.535	64.6043	325.81
<b>Mean</b>	<b>26.79</b>	<b>138.93</b>	<b>43.83</b>	<b>133.63</b>	<b>62.87</b>	<b>347.96</b>	<b>110.6</b>	<b>343.3</b>
<b>SD</b>	<b>9.97</b>	<b>9.79</b>	<b>24.46349736</b>	<b>31.23</b>	<b>24.57</b>	<b>33.19</b>	<b>70.21</b>	<b>76.5</b>

**Table S6. Real and virtual distances travelled by 12 participants in each technique and their corrected counterpart.**



**Figure S2. Mean of real and virtual distances travelled by 12 participants in each study along a path of 42 flags. Error bars represent SD.**



**Figure S3. Mean of real and virtual distances per flag travelled by 12 participants in each study. Error bars represent SD**