Multimodal Feedback and Interaction Techniques for Physically Based and Large Virtual Environments

Gabriel Cirio

To cite this version:

Gabriel Cirio. Multimodal Feedback and Interaction Techniques for Physically Based and Large Virtual Environments. Graphics [cs.GR]. INSA de Rennes, 2011. English. NNT : . tel-00652077

HAL Id: tel-00652077
https://theses.hal.science/tel-00652077

Submitted on 14 Dec 2011

Thèse

THÈSE INSA Rennes
sous le sceau de l'Université Européenne de Bretagne
pour obtenir le grade de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Informatique

présentée par
Gabriel CIRIO
ÉCOLE DOCTORALE : MATISSE
LABORATOIRE : INRIA Rennes - IRISA

Multimodal Feedback and Interaction Techniques for Physically Based and Large Virtual Environments

Thèse à soutenir le 2 Décembre 2011
devant le jury composé de :

Bruno Arnaldi
Professeur, INSA de Rennes / *Président*
Marie-Paule Cani
Professeur, Institut Polytechnique de Grenoble / *Rapporteuse*
Ming C. Lin
Professeur, University of North Carolina at Chapel Hill / *Rapporteuse*
George Drettakis
Directeur de Recherche, INRIA Sophia-Antipolis / *Examinateur*
Anthony Steed
Professeur, University College London / *Examinateur*
Anatole Lécuyer
Directeur de Recherche, INRIA Rennes / *Directeur de thèse*
Maud Marchal
Maître de Conférences, INSA de Rennes / *Encadrante*

Acknowledgments

Contents

Contents i

List of figures vii

List of tables xi

Introduction 1

I Haptic and Multimodal Interaction with Physically Based Complex Media 11

1 Related Work: Physically Based Haptic Interaction with Complex Media 13
- 1.1 Fundamentals of Haptic Interaction . . . 14
  - 1.1.1 Fundamentals of the Human Haptic System . . . 16
    - 1.1.1.1 Tactile receptors . . . 16
    - 1.1.1.2 Proprioceptive receptors . . . 17
    - 1.1.1.3 Force control . . . 17
  - 1.1.2 Haptic Devices . . . 17
    - 1.1.2.1 Tactile interfaces . . . 18
    - 1.1.2.2 Kinesthetic interfaces . . . 18
  - 1.1.3 Haptic Rendering . . . 19
    - 1.1.3.1 Closed-loop rendering . . . 20
    - 1.1.3.2 Open-loop rendering . . . 22
    - 1.1.3.3 Simulation loops . . . 23
- 1.2 Models for Physically Based Haptic Interaction . . . 23
  - 1.2.1 Rigid Bodies . . . 24
    - 1.2.1.1 Kinesthetic rendering . . . 25
    - 1.2.1.2 Vibrotactile rendering . . . 28
  - 1.2.2 Deformable Bodies . . . 29
    - 1.2.2.1 Kinesthetic rendering . . . 30
    - 1.2.2.2 Vibrotactile rendering . . . 35
  - 1.2.3 Fluids . . . 37
- 1.3 Combining Haptics with other modalities . . . 38
  - 1.3.1 Low-level integration . . . 39
  - 1.3.2 High-level integration . . . 40
- 1.4 Conclusion . . . 43

2 Six Degrees-of-Freedom Haptic Interaction with Fluids 45
- 2.1 Smoothed-Particle Hydrodynamics Fluid Simulation . . . 46
  - 2.1.1 SPH Discretization . . . 47

2.1.2 SPH Fluid Model . . . 47
2.2 3DoF Haptic Rendering of Fluids . . . 48
2.2.1 Smoothing Volume and SPH Haptic Forces . . . 48
2.2.2 Haptic Rendering Algorithm . . . 49
2.2.3 GPU Implementation . . . 49
2.2.3.1 Optimizations . . . 49
2.2.3.2 Update rate . . . 50
2.3 6DoF Haptic Rendering of Fluids through Rigid Body Interaction . . . 51
2.3.1 Unified Particle Model . . . 51
2.3.2 Rigid Body Dynamics . . . 53
2.3.3 6DoF Haptic Coupling Scheme . . . 54
2.3.4 Virtual Coupling . . . 55
2.4 Visual Fluid Rendering . . . 55
2.4.1 Computing Per-Pixel Fluid Data . . . 55
2.4.2 Fluid Compositing . . . 56
2.5 Evaluation . . . 56
2.5.1 Hardware Setup . . . 56
2.5.2 Computation Time . . . 57
2.5.2.1 Fluid Simulation Performance . . . 57
2.5.2.2 Unified Particle Model Performance . . . 57
2.5.3 Graphic Rendering . . . 58
2.5.4 Example Scenarios . . . 58
2.5.4.1 6DoF Interaction . . . 59
2.5.4.2 Container Interaction . . . 59
2.5.4.3 Variable Viscosity . . . 59
2.5.4.4 Bimanual coupling on the same rigid body . . . 60
2.5.5 A complete use-case . . . 61
2.6 Discussion . . . 61
2.7 Conclusion . . . 63

3 Six Degrees-of-Freedom Haptic Interaction with the Different States of Matter 65
3.1 SPH Multisate Simulation . . . 66
3.1.1 SPH Deformable Body Model . . . 67
3.1.2 Rigid Bodies, Fluids and Interaction Forces . . . 68
3.1.3 Changes of State . . . 68
3.1.4 Integration and Simulation Loop . . . 69
3.2 6DoF Multistate Haptic Rendering . . . 69
3.2.1 Rigid proxy . . . 69
3.2.2 Deformable proxy . . . 71
3.2.3 Proxy inducing state changes . . . 71
3.2.4 Friction Forces . . . 71
3.2.4.1 Contact point and surface detection . . . 72
3.2.4.2 Friction force computation . . . 72
3.3 Dual GPU Implementation . . . 73
3.4 Evaluation . . . 74
3.4.1 Haptic Time Step . . . 75
3.4.2 Haptic Feedback . . . 75
3.4.3 Perceptual Evaluation . . . 76

3.4.3.1 Population 76
3.4.3.2 Experimental Apparatus 76
3.4.3.3 Procedure 76
3.4.3.4 Experimental Plan 77
3.4.3.5 Results 77
3.4.3.6 Subjective Questionnaire 78
3.5 Discussion 78
3.6 Conclusion 80

4 Vibrotactile Rendering of Fluids **83**
4.1 Overview 84
4.2 Previous Approaches for Real-Time Fluid Sound Synthesis 84
4.3 Enhancing a Real-time Fluid Simulation with Bubbles 85
4.4 Vibrotactile Model 86
4.4.1 Initial Impact 86
4.4.1.1 Synthesis 86
4.4.1.2 Control 87
4.4.2 Harmonic Bubbles 87
4.4.2.1 Synthesis 87
4.4.2.2 Control 87
4.4.3 Main Cavity Oscillation 88
4.4.3.1 Synthesis 88
4.4.3.2 Control 88
4.5 Vibrotactile Rendering 89
4.6 Extension to Other Modalities 90
4.7 User Feedback 91
4.7.1 Scenario 91
4.7.2 Discussion 91
4.8 Conclusion 92

II Infinite Immersive Navigation Based on Natural Walking in Restricted Workspaces **93**

5 Related Work: 3D Walking Interfaces for the Navigation of Large Virtual Environments within Restricted Workspaces **95**
5.1 Locomotion Interfaces 96
5.1.1 Foot-based Devices 96
5.1.1.1 Foot-wearables 97
5.1.1.2 Foot platforms 97
5.1.2 Recentering Floors 98
5.1.2.1 Treadmills 98
5.1.2.2 Tiles 101
5.1.2.3 Spherical environments 101
5.2 3D Navigation techniques 102
5.2.1 Walking in place 103
5.2.2 Natural Walking Metaphors 103
5.2.3 Redirection Techniques 105
5.2.3.1 Redirected Walking 105
5.2.3.2 Motion Compression 107

5.2.3.3 Change Blindness 108
5.3 Conclusion 109

6 Infinite Navigation in Large Virtual Environments within Restricted Translation Workspaces 111
6.1 The Magic Barrier Tape 112
6.1.1 Display of the Workspace Limits 113
6.1.2 Navigation Through Rate Control 113
6.2 Extending Resetting Techniques for Omni-Directional Walking 115
6.2.1 Extended Freeze-Backup Technique. 115
6.2.2 Extended 2:1-Turn Technique. 116
6.3 Evaluation 116
6.3.1 Experiment #1: Pointing Task 116
6.3.1.1 Description 117
6.3.1.2 Results 118
6.3.2 Experiment #2: Path Following Task 119
6.3.2.1 Description 119
6.3.2.2 Results 121
6.3.3 Subjective Questionnaire 122
6.4 General Discussion 122
6.5 Conclusion 124

7 Infinite Navigation in Large Virtual Environments within Restricted Translation and Rotation Workspaces 127
7.1 Three Novel Navigation Techniques 128
7.1.1 Motivation for New Navigation Metaphors 129
7.1.2 General Terminology and Quantities 129
7.1.3 Constrained Wand and Signs 130
7.1.4 Extended Magic Barrier Tape 131
7.1.5 Virtual Companion 132
7.2 Evaluation 133
7.2.1 Experimental Conditions 134
7.2.1.1 Population 134
7.2.1.2 Experimental Apparatus 134
7.2.1.3 Procedure 135
7.2.1.4 Collected data 135
7.2.2 Task #1: Pointing Task 135
7.2.3 Task #2: Path Following Task 136
7.3 Results 136
7.3.1 Recorded time and tracking data 136
7.3.1.1 Time to complete the task 136
7.3.1.2 Physical walking distance 137
7.3.1.3 Time spent in reaction and danger zones 137
7.3.1.4 Deviation from the ideal path 138
7.3.1.5 VE differences 138
7.3.2 Questionnaire 138
7.4 Discussion 139
7.5 Conclusion 141

Conclusion 143

A Appendix: Fundamentals of Physically Based Simulation Models for Haptic Interaction 151
A.1 Rigid bodies . . . 151
A.2 Deformable bodies . . . 153
A.2.1 Continuum mechanics . . . 153
A.2.2 The Finite Element Method . . . 154
A.2.2.1 Linear FEM . . . 156
A.2.2.2 Corotated formulation . . . 156
A.2.3 Mass-spring systems . . . 156
A.3 Fluids . . . 157
A.3.1 Navier-Stokes equations . . . 157
A.3.2 Eulerian simulation . . . 158
A.3.2.1 Extensions . . . 160
A.4 Time integration schemes . . . 161
A.4.1 Explicit integration . . . 162
A.4.1.1 The explicit Euler method . . . 162
A.4.1.2 The Runge-Kutta method . . . 162
A.4.2 The implicit Euler method . . . 163

B Appendix: Résumé Long en Français 165
B.1 Partie 1: Interaction Haptique et Multimodale avec des Milieux Complexes Basés Physique . . . 169
B.1.1 Interaction haptique à 6DDL avec des fluides . . . 169
B.1.1.1 Simulation physique de fluides . . . 169
B.1.1.2 Simulation d'objets rigides . . . 169
B.1.1.3 Rendu haptique à 6DDL . . . 170
B.1.1.4 Scénarios de test . . . 171
B.1.2 Interaction haptique à 6DDL avec les différents états de la matière . . . 171
B.1.2.1 Simulation d'objets déformables . . . 172
B.1.2.2 Rendu haptique multi-état à 6DDL . . . 172
B.1.2.3 Évaluation . . . 172
B.1.3 Interaction vibrotactile et multimodale avec des fluides . . . 173
B.1.3.1 Simulation SPH de bulles . . . 174
B.1.3.2 Modèle vibrotactile . . . 174
B.1.3.3 Rendu vibrotactile et multimodal . . . 175
B.2 Partie 2: Navigation Immersive et Infinie Basée sur la Marche dans des Espaces de Travail Restreints . . . 176
B.2.1 Navigation infinie d'EV dans des espaces de travail restreints en translation 176
B.2.1.1 Le "Bandeau Magique" . . . 176
B.2.1.2 Évaluation . . . 177
B.2.2 Navigation infinie d'EV dans des espaces de travail restreints en translation et en rotation . . . 177
B.2.2.1 Trois nouvelles techniques de navigation . . . 177
B.2.2.2 Évaluation . . . 178
B.3 Conclusion . . . 179

Publications 181

Bibliography 181

List of Figures

1 Concept: a person walking on a beach 2
2 Objectives of Axis 1 5
3 Objectives of Axis 2 6

1.1 Architecture of a VR application 14
1.2 Human and machine haptic loops 15
1.3 Examples of tactile interfaces 19
1.4 Examples of kinesthetic interfaces 20
1.5 Closed-loop rendering 20
1.6 Virtual coupling mechanism 22
1.7 Open-loop rendering 22
1.8 6DoF God-object technique 26
1.9 Voxel-based rendering discretizations 27
1.10 Example of voxel-based rendering 28
1.11 Vibrotactile rendering of rigid contacts 29
1.12 Spatialized haptic rendering 30
1.13 Non-linear deformations using mass-spring systems 31
1.14 Voxel-based rendering of deformable bodies 33
1.15 Example of voxel-based rendering of highly detailed deformable bodies 33
1.16 Linear deformations using an LCP formulation 34
1.17 Non-linear deformations using an LCP formulation 34
1.18 Meshless rendering 35
1.19 Fracture mechanics for the vibrotactile rendering of granular materials 36
1.20 Examples of vibrotactile rendering of granular materials 36
1.21 Precomputed haptic interaction with fluids 38
1.22 Physically based haptic interaction with viscous fluid 39
1.23 Physically based haptic interaction with smoke 39
1.24 The frozen pond 40
1.25 High-level multimodal integration for virtual prototyping 41
1.26 Elaborate high-level multimodal integration 42
1.27 The Munich Knee Joint Simulation project 43

2.1 Smoothing Volume and SPH haptic forces 48
2.2 Overview of the 6DoF haptic rendering of fluids 51
2.3 Rigid body particle sampling 52
2.4 Illustration of the computation of forces acting on a rigid body 53
2.5 The bilateral sampling kernel 55
2.6 Performance evaluation of our fluid simulation algorithms 57
2.7 Evaluation of our graphic rendering method 58
2.8 6DoF haptic interaction scenario 59
2.9 Container interaction scenario 60

2.10 Variable viscosity scenario 60
2.11 Bimanual coupling on the same rigid body 61
2.12 A user with the virtual crepe factory 61
2.13 New interaction possibilities with the virtual crepe factory 62

3.1 The states of matter in a physical simulation 66
3.2 6DoF multistate haptic rendering 71
3.3 Contact point and surface detection on a proxy rigid body 72
3.4 Illustration of the decoupled GPU architecture 74
3.5 Haptic time step of a multimodal VE 75
3.6 Haptic feedback profile for a multistate VE 76
3.7 Experimental apparatus of the multistate evaluation 77
3.8 The three different states of matter of the multistate evaluation 77
3.9 Results for subjective ratings in the multistate evaluation 79
3.10 Scenario illustrating a cooking simulator 79
3.11 Scenario illustrating the changes of state 80

4.1 Overview of our vibrotactile fluid approach 85
4.2 The three components of our vibrotactile model 86
4.3 Interaction examples for the vibrotactile rendering of fluids 89
4.4 Vibrotactile signal generated with our model 90

5.1 Foot-wearable devices 97
5.2 Foot platforms 98
5.3 1DoF treadmills 100
5.4 1DoF treadmills for dynamic terrain 100
5.5 Omni-directional treadmills 101
5.6 The CirculaFloor 101
5.7 Spherical environments 102
5.8 The Step WIM 104
5.9 The VE used to test the Redirected Walking technique 106
5.10 The VE used to test passive haptics with the Redirected Walking technique . . 107
5.11 Motion Compression real and virtual paths 108
5.12 Another example of Motion Compression 108
5.13 Sequence of changes exploiting change blindness 109

6.1 The Magic Barrier Tape 112
6.2 The three Magic Barrier Tape visual cues 114
6.3 The Gaussian deformation of the Magic Barrier Tape 115
6.4 The visual cues from the extended resetting techniques 116
6.5 A subject wearing the tracking equipment 117
6.6 VE used in Experiment #1 of the Magic Barrier Tape 118
6.7 Results of Experiment #1 of the Magic Barrier Tape 119
6.8 Two paths used in Experiment #2 of the Magic Barrier Tape 120
6.9 The VE used in the Experiment #2 of the Magic Barrier Tape 120
6.10 Results of Experiment #2 of the Magic Barrier Tape 122
6.11 Questionnaire results of the Magic Barrier Tape 123

7.1 Screenshots illustrating the three techniques 128
7.2 Regions and boundaries for translation and rotation 130

7.3 The constrained wand . . . 131
7.4 The extended MBT . . . 132
7.5 The Virtual Companion . . . 133
7.6 The gesture set for controlling the Virtual Companion . . . 133
7.7 The simple VE used for the first block of tests . . . 134
7.8 Recorded trajectories of a participant . . . 136
7.9 Equivalence groups for the recorded time and tracking data . . . 137
7.10 Questionnaire results . . . 139

List of Tables

2.1 Comparison of our haptic fluid approach with previous work 62

3.1 Probabilities of correct answers for each state of matter 78

7.1 Summary overview of the main advantages and drawbacks of our techniques. . 140

Introduction

This Ph.D. manuscript, entitled “Multimodal Feedback and Interaction Techniques for Physically Based and Large Virtual Environments”, presents research conducted in the context of Virtual Reality (VR). VR technologies aim at simulating digital environments with which users can interact and, as a result, perceive through different modalities the effects of their actions in real time. Burdea and Coiffet [1] define VR as *“a high-end user-computer interface that involves real-time simulation and interactions through multiple sensory channels. These sensory modalities are visual, tactile, auditory, smell, and taste”*.

VR has the inherent capacity to realistically create and simulate specific virtual environments (VE), even before these environments are actually used or even built in real life. Furthermore, VR is not limited to copying and imitating real world scenarios and behaviors: it allows the creation and simulation of any sort of VE, limited only by the imagination of the designer and the capabilities of the system. As a consequence, VR can be found in different applied domains outside of the many research labs devoted to this field.

In the automotive industry, for instance, a vehicle in the design stage can be displayed through VR, allowing the identification of potential design problems without producing an expensive physical mock-up. Virtual assembly and maintenance procedures can help validate or modify real procedures [2], while workforces can be trained through VR scenarios, greatly reducing training costs and risks [3]. VR has also drawn the attention of the medical field, with multimodal simulators for the training of surgeons [4, 5], dentists [6] and orthopedists [7, 8]. Patients suffering from phobias can be immersed into VE [9] in order to treat the pathology in a completely safe and controlled environment. Other application areas of VR include the fields of entertainment (video games and motion simulators), education (enhanced visualization, distant learning, virtual museums, sports) and design (CAD, architectural mockups, virtual art).

Unfortunately, in the current state of the art there are many limitations in terms of interaction possibilities in VR, both in available hardware and software components. Many of these limitations arise when interacting with complex VE. For instance, it is quite challenging to simulate natural phenomena under VR constraints, namely in real-time and with high quality feedback. It is also very hard to design a device allowing a user to walk without real forward motion while providing an accurate restitution of walking sensations. The main limiting factors are the available computational power, the limited technology, and the inherently complex nature of natural phenomena. In fact, most situations that we live in our (real) life cannot be simulated in a faithful manner. For instance, it is not yet possible to faithfully simulate the multimodal exploration of natural scenes such as the one illustrated in Figure 1, i.e. walking on a beach. Water motion and sand compliance are complex phenomena, and although there are physically based models to simulate them, real-time constraints and multimodal feedback present a considerable challenge. Walking naturally on the scene to reach to other end of the beach is just impossible, due to limited tracking areas and the minuscular size of immersive environments compared to natural landscapes.

Figure 1 – A person walking on a beach, interacting with water, sand and stones, while traveling along the beach shore. This scenario cannot be efficiently simulated in VR nowadays, due to the limited availability of multistate simulations and multimodal feedback models, and due to the boundaries of the real workspace.

Research Context

Interaction significantly contributes in making VR such a powerful and immersive tool. The more believable the interaction and its feedback, the more it makes the user unconsciously shift his reality from the real to the virtual environment, developing a true sense of presence [1]. Taking into account the capabilities of today's and tomorrow's VR systems, we define the research context of this thesis based on three fundamental conditions for VR interaction[2]. This context will allow us to highlighting the main weaknesses of current approaches, and point out what aspects have remained largely unexplored. These issues will lead us to define our different research axes and, most importantly, drive our work.

- **Interaction in VR should be multimodal**. In real life, we interact with our surrounding environment with our five senses. Each sense provides complementary cues for a wider and more accurate perception. Ideally, it should be the same in a VR simulation. It should be safe to state that, for most tasks, humans rely on vision, hearing and touch. Thus, we believe these three modalities should be simulated and rendered to the user in immersive VR applications.

- **Interaction in VR should be physically based**. Users expect the VE to behave like in the real world, except for very specific scenarios. Objects are supposed

---

[1] *Presence* can be defined as the illusion of being located inside the VE depicted by the VR system: the "sense of being there" [10]

[2] Not following these conditions does not necessarily mean that VR interaction is not possible. Many uses of VR do not require the fulfillment of these three conditions.

to fall, collide, deform and flow as usual, and should respond to user actions with realistic behavior. Thus, they have to follow the different laws of physics, at least from a macroscopic point of view. Doing this geometrically or by predefined animation keys only works for specific, precomputed, and therefore limited scenarios. For full interaction possibilities with different VE, the behavior has to be described by physically based models of the different objects populating the scene.

- **Interaction in VR should allow complex VE**. By complex, we refer to a higher demand in the characteristics of the VE and/or its objects. There are many ways in which a VE can be complex:
  - in size: large objects, large scenes
  - in number: high polygon count, high object count
  - in shape: small-scale details, convex objects, landscape with relief
  - in behavior: non-rigid media (deformable bodies, fluids), large dynamic components (speed, force)

  Users should be able to interact with complex environments, as they actually represent most real-life scenarios. However, since they are labeled as complex, they inherently pose computation, modeling or interface challenges.

When enforcing these three conditions in a VR simulation, we are often confronted to the issues discussed earlier, namely limited available computational power, limited technology, and the inherently complex nature of physical phenomena. Thus, in this Ph.D. thesis, we focus on enhancing the multimodal and physically based interaction with complex VE.

In order to address this problem, we adopt a subdivision approach by breaking it down into two subproblems, following the main categories of VR interaction techniques. As defined by Hinckley *et al.* [11], *"an interaction technique is the fusion of input and output, consisting of all software and hardware elements, that provides a way for the user to accomplish a task"*. From Bowman *et al.* [12] seminal taxonomy of VR tasks, we focus on two main categories that can be identified within VR interaction techniques [3]:

- the *manipulation* category (and the very related *selection* category), regrouping the interaction techniques allowing the user to interact with the objects constituting the VE,
- the *navigation* category, regrouping the interaction techniques allowing the user to move within the VE.

These categories represent the tasks that could be performed by a user in a real environment.

Objectives

In this Ph.D. thesis we focused on two research axes, corresponding to the multimodal and physically based interaction with complex VE within both fundamental interaction

---

[3] We do not consider system control and symbolic input tasks, since these are a response to user interface issues and do not arise from real world tasks transposed to VR.

categories: manipulation (or interaction with virtual objects) and navigation. The word “complex” implies different challenges according to each category, and thus raises different conceptual and technical issues. Therefore, the main objective of this thesis is formulated as more specific objectives under the research axis corresponding to each category.

Axis 1 - Multimodal *manipulation* of physically based complex VE

The addition of haptic kinesthetic (force) feedback is a step forward to fully multimodal simulations. Touching, feeling objects when manipulating them is in our very nature, as virtually all tasks we accomplish in real life involve bodily interaction with the environment. It is in fact quite likely that a higher sense of presence could be generated in a VR simulation by adding a simple force feedback interface with low resolution force restitution to an existing visual and auditory VR setup, than by improving one particular modality such as the visual display alone [13]. Besides, the addition of force feedback to VR simulations has been shown to improve user immersion and performance [14, 15] when accomplishing some specific tasks in the VE. Vibrotactile and acoustic feedback are also attractive additions to VR simulations, since they do not require expensive robotic devices as other modalities: there is a wide availability of off-the-shelf and easily built vibrotactile hardware (actuated floors [16], shoes [17], and hand-held transducers) and acoustic devices (speakers).

First objective: Multimodal manipulation of fluids

Most current multimodal simulations involve only rigid bodies, since they follow simple dynamics and represent many of the objects that surround us. Complex VE, however, can have non-rigid media, exhibiting many more degrees of freedom, and following more complex phenomena. Different physically based approaches have been developed for the real-time simulation of non-rigid media such as elastic bodies and fluids. However, the multimodal interaction with these media has room for improvement, and is limited nowadays by the available computational power. Multimodal interaction with deformable bodies have received some attention in the context of force feedback. Surprisingly, multimodal interaction with fluids has been scarcely studied. However, we often interact with fluids in our daily life, either through tools such as when holding a glass of water or stepping on a puddle with our shoes, or directly with our body when we swim, wash our hands or walk on a beach shore. Fluids are also found in many applications such as for industrial or medical manipulations - involving for instance blood flow and natural liquids. Water, an example of fluid, is the most manipulated material [18] in industry. Enabling multimodal feedback in the interaction with fluids, besides allowing more realistic simulations, would enable a wide range of novel simulation scenarios and applications.

Second objective: A unified approach for the manipulation of media with force feedback

Complex VE with fluid inside are usually also populated by solid (rigid and deformable) media. Generally speaking, complex VE usually involve simulating several types of media at the same time. However, simulating fluid, deformable and rigid media in the same simulation with haptic feedback poses several challenges. What should be a fairly common scenario implies the simulation of heterogeneous media through different models specific to each medium. Their interactions have to be computed, thus requiring coupling mechanisms between each couple of media. And, most importantly, the user needs to interact with the VE and receive force feedback, thus requiring haptic coupling mechanisms for each medium present in the VE. Taking into account these constraints leads to an increase in the complexity and the computational cost of an already highly complex and time consuming simulation. Previous haptic rendering techniques focus on a single medium, and existing multistate approaches are quite limited [19]. Haptic interaction with different physically based media would be more efficient and seamless to users, designers and developers through a unified approach for simulation and rendering.

Figure 2 illustrates the objectives of Axis 1: the multimodal manipulation of complex VE with rigid, deformable and fluid media through different modalities (kiensthetic, tactile, acoustic and visual).

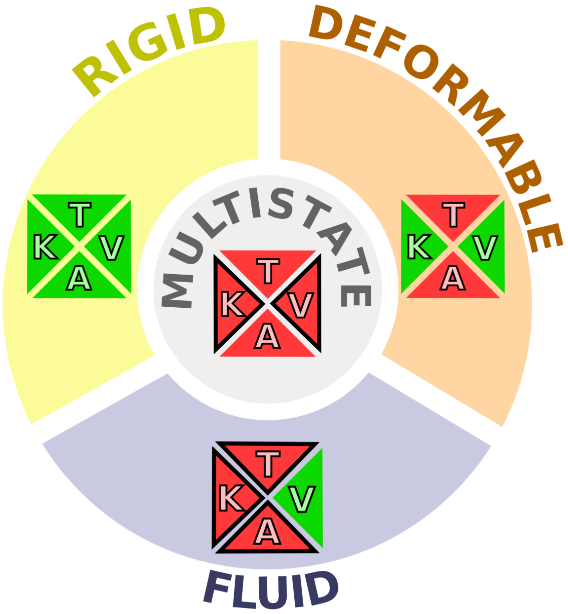


Figure 2 – Objectives of Axis 1: multimodal manipulation of complex VE. Modalities: K (kinesthetic), T (tactile), A (acoustic), V (visual). Modalities that have been largely addressed in previous work are shown in green. Modalities that have been scarcely studied or not studied at all are shown in red. Those with a black frame are addressed in this manuscript.

Axis 2 - Multimodal *navigation* of physically based complex VE

There is a wide range of devices and metaphors for the navigation of VE. Following the fundamental conditions of our research context, we require a navigation interface providing multimodal feedback. Instead of relying on computationally expensive artificially generated sensory feedback, we simply focus on natural walking as the core of the navigation interface. Indeed, using natural walking in a VE inherently matches vestibular and proprioceptive cues from the real movement, but with the visual feedback from the virtual movement. Natural walking also naturally produces vibrotactile and acoustic feedback when stepping on the real ground. Thus, natural walking in a VE produces a perfectly accurate multi-sensory perception of navigation, hard to match with simulated approaches. It also provides the most natural, intuitive and direct way of controlling one's position. In addition, several studies have shown the benefits of using natural walking for the navigation of VE, in terms of task performance [20, 21, 22], presence [23] and naturalness [24, 23, 22]. Besides, natural walking is, after all, the locomotion interface that we use in our everyday life.

Third objective: immersive navigation techniques for large VE based on natural walking

VE span from small rooms where everything is accessible under one's arm reach, to complex very large environments representing, for instance, outdoor scenes. Even infinitely large scenes, although non existent in real life but relatively easy to implement, are virtually possible. Although the design of a large environment is not necessarily complex (it actually depends on how it is populated), navigating it does pose many challenges: the VE might be very large or even infinite, but the real physical workspace is not. In most cases, the space in which the user moves is significantly smaller than the simulated environment. This is the case for CAVE-like setups where the 4 screens represent the boundaries of the workspace, but it also applies to HMD setups since workspaces are bounded by the range of tracking systems. There are also boundaries rotation-wise in CAVE-like setups, where one screen (the "back" screen) is missing. In any case, walking users eventually reach the boundaries of the workspace, leading to breaks of immersion, blocking situations and safety problems. Providing an immersive and safe walking metaphor for navigating in infinite VE within the confines of restricted workspaces is a challenging task. It would provide a solution to many training and entertainment VR simulations requiring large scenes.

Figure 3 illustrates the objective of Axis 2: the multimodal navigation of complex VE, when the VE is larger than the available physical workspace.

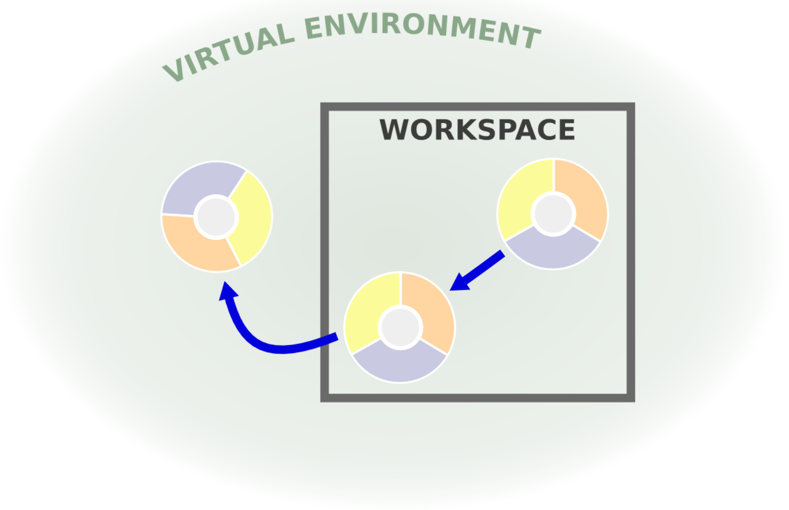


Figure 3 – Objectives of Axis 2: multimodal navigation of complex VE. The black square represents the boundaries of the workspace. Multimodal interaction can happen in the VE inside the boundaries, but raise several issues when moving outside of the workspace, in translation and rotation.

Approach and Contributions

This manuscript presents the research carried out in order to address the three objectives mentioned above. It is naturally divided in two parts, each following a research axis:

- Part I describes novel techniques for **haptic and multimodal interaction with physically based and complex media**. Haptic feedback includes kinesthetic (force) feedback as well as vibrotactile feedback. These modalities are combined to visual and, to some extent, acoustic feedback. The focus is given to fluids, since these are scarcely explored in previous work while being widely present in real life and in different VR application fields. Rigid and deformable bodies are also considered when proposing a unified approach for the kinesthetic interaction with different media.

- Part II describes novel metaphors for **infinite immersive navigation based on natural walking in restricted workspaces**. Both translational boundaries (screens, tracking range) and rotational boundaries (missing screens) are considered in the design of these metaphors.

More details are given in the remainder of this chapter.

Part 1 - Haptic and Multimodal Interaction with Physically Based Complex Media

We first propose a background overview of haptic and multimodal interaction with physically based complex media in **Chapter 1**. We begin by providing an introduction to the **fundamentals of haptic interaction**, namely the human haptic system, the haptic devices and the main concepts of haptic rendering. Then, we focus on the **existing physically based models for haptic interaction for the different media** of the computer graphics field: rigid bodies, deformable bodies and fluids. Finally, we provide an overview of **existing multimodal approaches** in different application areas involving the combination of visual and haptic feedback with other modalities.

The haptic interactive simulation of fluids is particularly challenging, especially to achieve realistic and stable force-feedback with high update rates using physically based models. To simulate interactions between fluids and rigid bodies with haptic rendering, previous studies have proposed precomputed ad-hoc algorithms [25], approaches featuring only 3 Degrees of Freedom (DoF) and non-viscous fluids [26], or implementations restricted to simple object shapes and small amounts of fluid [27]. Thus, as for today, there is a lack of models and rendering techniques handling complex 6DoF haptic interactions with viscous fluids in real-time.

In **Chapter 2** we propose a novel approach that allows **real-time 6 Degrees of Freedom haptic interaction with fluids of variable viscosity**. Our haptic rendering technique is based on the Smoothed-Particle Hydrodynamics [28, 29] model, and uses a **new haptic coupling scheme** and a **unified particle model** allowing the use of arbitrary-shaped rigid bodies. Particularly, fluid containers can be created to hold fluid and hence transmit to the user force feedback coming from fluid stirring, pouring, shaking and scooping, to name a few. In addition, we **adapted an existing visual rendering algorithm** to meet the frame rate requirements of the haptic algorithms. We **evaluate and illustrate** the main features of our approach through different scenarios, highlighting the 6DoF haptic feedback and the use of containers.

When populating the VE with multiple states of matter (fluid, deformable and rigid media), the complexity of the simulation increases significantly. Different simulation models have to coexist and interact within the same simulation, while running together. When providing haptic feedback, haptic coupling mechanisms and rendering techniques have to take into account the specificities of each medium. Existing haptic rendering techniques allow the physically based interaction with at most two different states of matter. Thankfully, in the computer graphics field Smoothed-Particle Hydrodynamics have been used for the offline simulation of different media within the same framework.

Hence, in **Chapter 3** we naturally extend the haptic fluid approach presented in Chapter 2 to include deformable bodies, thus introducing the first approach that provides a **unified physically based haptic feedback for fluid, deformable and rigid states of matter** in the same simulation. This method avoids the complexity of dealing with different simulation algorithms and their coupling, and uses a single haptic rendering mechanism. The approach is enhanced with **state change mechanisms, friction forces and multistate proxies**. Haptic rates are achieved through a **dual GPU implementation**. The approach is **evaluated by assessing the capability of users to recognize the different states of matter** they interact with.

Force and visual feedback are not the only important modalities when interacting with the environment. Vibrotactile and acoustic feedback provide complementary cues for a better perception of materials, forces and distances, among others. The availability of cheap transducers for these modalities makes them an interesting addition to VR applications. Indeed, many common materials with which we interact on a daily basis can be simulated and displayed through the vibrotactile and acoustic modalities in real-time. Examples include solids such as wood and metal [30] and aggregates such as gravel and snow [31, 32]. However, materials such as water and other fluids have again been largely ignored in this context. Compelling multimodal VR simulations such as walking through puddles or splashing on the beach are very limited without these additional cues, but would be of great interest in the entertainment field.

To this end, in **Chapter 4** we introduce the first approach for the **vibrotactile rendering of fluids**. Similar to other rendering approaches for virtual materials [31, 32, 30], we leverage the fact that vibrotactile and acoustic phenomena share a common physical source. Hence, we base the design of our vibrotactile model on **prior knowledge of fluid sound rendering**. Since fluid sound is generated mainly through bubble and air cavity resonance, we enhanced our fluid simulator presented in Chapter 2 with **real-time bubble creation and solid-fluid impact mechanisms**. We can synthesize vibrotactile feedback, and to some extent acoustic feedback, from interaction and simulation events. Using this approach, we are exploring the use of bubble-based vibrations to convey fluid interaction sensations to users. We render the feedback for hand-based and, in a more innovative way, for **foot-based interaction**, engendering a rich perceptual experience of feeling the sensations of water.

Part 2 - Infinite Immersive Navigation Based on Natural Walking in Restricted Workspaces

We begin the second part of this manuscript by surveying in **Chapter 5** the existing 3D user interfaces using walking for the navigation of large environments within the confines of restricted workspaces. We first review **existing locomotion interfaces**, which propose hardware solutions. We study foot-based devices, which compensate the motion of each foot separately, and recentering floors, which compensate the overall movement. Then, we focus on software solutions with the **existing 3D navigation techniques**.

These include walking in place approaches, where the user performs the walking gait but without forward motion, natural walking metaphors, which combine natural walking with conscious and complementary techniques for dealing with the boundaries, and redirection techniques, which trick the user into modifying his trajectory in the VE.

When navigating in large or infinite VE within workspaces with restricted size, users are faced to the problem of reaching the workspace boundaries, thus raising safety problems and breaking immersion if not properly addressed. There are hardware and software-based approaches to overcome these issues: locomotion interfaces [33] such as treadmills often have major limitations that restrict their widespread use (huge size and weight, high cost, lack of accuracy), while existing navigation techniques [10, 34, 35, 36] often fail at providing a simple, intuitive and immersive interaction.

Therefore, in **Chapter 6** we introduce a **novel interaction metaphor called the Magic Barrier Tape**, which allows a user to **navigate in a potentially infinite VE** while confined to walking workspaces restricted in translation. Head-Mounted Displays (HMD) with limited tracking range are examples of such workspaces. The technique relies on the barrier tape metaphor and its “do not cross” implicit message by surrounding the walking workspace with a virtual barrier tape in the VE. Therefore, the technique informs the user about the boundaries of his walking workspace, providing an environment safe from collisions and tracking problems. It uses a **hybrid position/rate control** mechanism to enable natural walking inside the workspace and rate control navigation to move beyond the boundaries by “pushing” on the virtual barrier tape. It provides an **easy, intuitive and safe way of navigating** in a VE, without break of immersion. Two experiments were conducted in order to **evaluate** the Magic Barrier Tape by comparing it to two navigation techniques sharing the same objectives.

The issue of reaching the boundaries of the workspace appears not only when the user moves towards the boundaries in translation: it can also happen when the user moves in rotation. Some VR setups, such as CAVE-like environments, present additional workspace restrictions. Indeed, in these setups users are not immersed in 360°: there are missing screens, leading to breaks of immersion when noticed by the user while turning. Hence, some workspaces are limited in translation *and* rotation.

Chapter 7 presents **three new techniques that deal with translation and rotation issues through common metaphors**. These techniques provide a navigation metaphor that keeps the user safe from the boundaries, without breaking immersion. The first metaphor extends the basic and well-known wand paradigm by adding **virtual warning signs**. The second metaphor extends the Magic Barrier Tape presented in Chapter 6 by adding **virtual walls** that prevent the user from looking at the missing screen. The third metaphor introduces a **virtual companion** in the form of a bird to guide and protect the user within the VE. These techniques are **evaluated** by comparing them first to a base wand condition. The study provides insight into the relative strengths of each new technique, while showing that they can efficiently address the issues of navigation in large VE within restricted workspaces.

Finally, **Chapter 8** provides **conclusions and perspectives** of the work presented in this manuscript.

Part I

Haptic and Multimodal Interaction with Physically Based Complex Media

Related Work: Physically Based Haptic Interaction with Complex Media

1

Contents

1.1 Fundamentals of Haptic Interaction 14
  1.1.1 Fundamentals of the Human Haptic System 16
  1.1.2 Haptic Devices 17
  1.1.3 Haptic Rendering 19
1.2 Models for Physically Based Haptic Interaction 23
  1.2.1 Rigid Bodies 24
  1.2.2 Deformable Bodies 29
  1.2.3 Fluids 37
1.3 Combining Haptics with other modalities 38
  1.3.1 Low-level integration 39
  1.3.2 High-level integration 40
1.4 Conclusion 43

Just as the synthesizing and rendering of visual images defines the area of computer graphics, the art and science of developing devices and algorithms that synthesize computer generated force-feedback and vibrotactile cues is the concern of computer haptics [37]. Haptics broadly refers to touch interactions (physical contact) that occur for the purpose of perception or manipulation of objects [38].

In this chapter, we survey previous work on haptic interaction with physically based VE. These environments are often populated with complex media, such as detailed rigid bodies, deformable objects and volumes of fluid. Thus, a significant number of approaches have been developed since the introduction of haptics, allowing the haptic interaction with different bodies of different media. We first recall the fundamentals of haptic interaction, focusing on the main mechanisms behind human, device and software haptic components. Then, we survey the existing physically based models for haptic interaction, allowing the computation of force or vibrotactile feedback from the interaction with objects in the rigid, deformable and fluid states of matter. Finally, we provide an overview of existing multimodal approaches in different application areas involving the combination of visual and haptic feedback with other modalities.

1.1 Fundamentals of Haptic Interaction

Haptics have recently started to be included in VR setups and applications alongside the visual and acoustic modalities. Figure 1.1 shows the structure of a VR application with visual and haptic feedback [38]. A simulation engine uses one or several models to compute the VE behavior through time. The engine generates data, such as positions, velocities and forces, which are sent to visual and haptic rendering algorithms. These algorithms are responsible for the computation of the VE graphic and force/tactile signals conveying the different cues to the user. These signals are fed to transducers, which are the physical devices responsible for the conversion of signals into physical stimuli in visual or tactile form, perceived by the user: a new position of the haptic device handle, a specific vibration of a wearable vibrator, and an image drawn on a screen, for example.

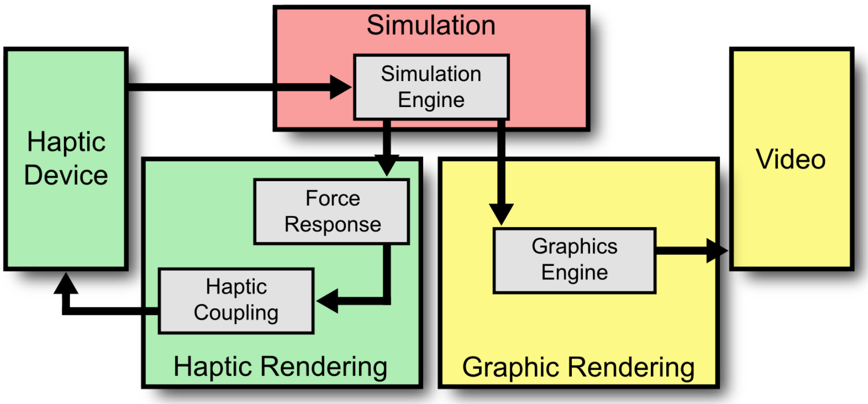


Figure 1.1 – Architecture of a VR application with visual and haptic modalities. Inspired from [38]

.

A specificity of haptics compared to others sensory modalities is its bidirectional flow. When subject to kinesthetic rendering, a user perceives a force *from* the haptic device, but also exerts a force *on* the haptic device. This is not the case with audio or visual modalities, where a user does not affect the rendering device and hence the sensory loop. This bi-directionality, the capacity to exchange information and energy in two directions, from and toward the user, is often referred to as the single most important feature of the haptic modality [38], and clearly highlights its interactive nature.

The interest for force and touch feedback in computer simulations goes back to the mid-sixties [39, 40, 41]. Since then, it has been shown that haptic feedback enhances the immersion of users in VR, as well as their performance in the achievement of a task within a VE [14, 15, 42]. With the fast growing in computational power and in device rendering fidelity, many areas have been targeted by past and present applications of ongoing research, and exciting possibilities can be foreseen in the near future [13]:

- Medicine: surgical simulators for medical training, remote diagnosis for telemedicine, aids for the blind such as warning or path guidance, rehabilitation of patients with gait problems
- Industry: path planning, virtual prototyping, virtual assembly, virtual training
- Scientific visualization: exploration of complex data sets, molecular manipulation

- Entertainment: video games and simulators for a deeper immersion within the VE
- Exhibitions: virtual art, virtual touching in museums
- Content creation: enhanced modeling and virtual sculpturing, 3D painting
- Architecture and design: virtual walkthrough, model testing

As defined by Srinivasan and Basdogan [13], research in the area of haptics can be categorized into two main areas: Human Haptics and Machine Haptics. These categories are tightly linked to the subsystems and information flow behind the haptic interaction between a human user and the VE through the haptic interface. Human haptics are related to the human sensorimotor loop: when a human user touches a real or virtual object, forces or vibrations are exerted on the skin and the muscles. The information is sensed by different receptors, depending on the type of stimuli. The associated sensory information is conveyed to the brain and leads to conscious or unconscious perception. The brain issues motor commands to activate the different effectors, eventually resulting in motion. Conversely, Machine Haptics are related to the Machine sensorimotor loop: when the human user manipulates the haptic device, the device sensors convey the different sensed data to the computer. The VE is updated, and the computed output data is sent to the actuators of the haptic device to generate the haptic feedback. Both categories form two distinct loops in the haptic interaction process, as show in Figure 1.2.

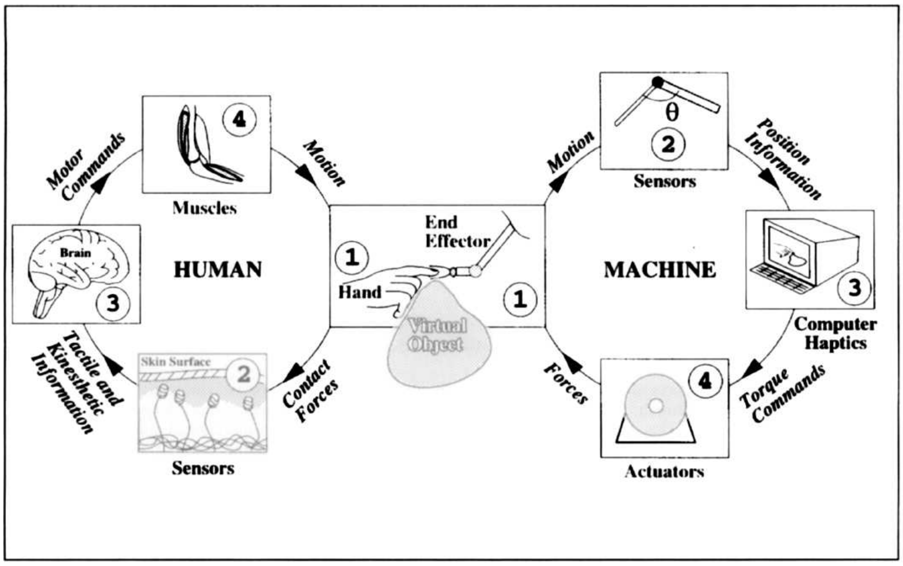


Figure 1.2 – Human and machine loops during haptic interaction [13]

Within the machine haptics area, the haptic rendering refers to the process by which sensory stimuli are computed through a software algorithm in order to convey information about a virtual object [38]. A haptic rendering algorithm gathers data from the environment, such as the device position and the physical attributes of the virtual objects (shape, elasticity, texture, mass, etc), and produces force, torque and/or tactile signals. The design of the algorithm is crucial for an accurate stimuli restitution. It is analogous to a graphic rendering algorithm: a sphere visually rendered with simple shading techniques will look different from the same sphere rendered with ray-tracing techniques. In haptics, a sphere rendered using simple geometrical functions will feel different from the same sphere rendered with physically based techniques conveying texture and friction sensations [38].

1.1.1 Fundamentals of the Human Haptic System

The human haptic system consists of the mechanical, sensory, motor and cognitive components of the hand-brain system [13]. Here, we overview the main receptors involved in haptic perception and their roles, and provide some performance information related to the haptic modality. This information plays an important role when designing feedback mechanisms, since output signals can take into account the different sensory thresholds and thus maximize the efficiency and quality of the restitution.

Receptors of the human haptic system are located mainly in the skin, joints, tendons, and muscles. These receptors are activated through stimuli of different type (mechanical, thermal or chemical). Information collected by receptors is conveyed to the central nervous system, mainly the brain and the spinal cord, using electrical impulses through the afferent (sensory) neural network. The responses generated in the central nervous system travel through the efferent (motor) nerve fibers, conducting impulses to motor neurons that transmute neural signals into activation of muscles and glands [13].

Haptic sensory information from the body in contact with an object can be divided into two classes: *tactile perception* involving the perception of sensations at the surface of the skin, and *kinesthetic (proprioceptive) perception* involving the perception of body positions and forces [43]. Upon contact, forces are usually sensed by both tactile and kinesthetic systems. Coarse properties of objects explored through hand or arm motion, such as large shapes (one meter or more) or spring-like compliances, are conveyed by the kinesthetic system. On the other hand, spatiotemporal variations of contact forces are usually sensed by the tactile system, including fine shapes, texture, slip, and rubber-like compliances, among others [13].

1.1.1.1 Tactile receptors

Tactile sensations result from the stimulation of three kind of receptors located in the skin [13]:

- *thermoreceptors*, sensitive to temperature. There are two types of thermoreceptors, sensitive to changes in cold or warm temperatures.
- *nociceptors*, sensitive to mechanical, thermal or chemical stimuli that have the potential to damage tissues.
- *mechanoreceptors*, composed of different receptors sensitive to mechanical stimulations like pressure, vibrations, flutter, stretch and textures. Among the four types of mechanoreceptors, we can distinguish: 1) *slow adapting receptors* which are stimulated throughout a sustained stimulus (Merkel disks, sensitive to unchanging pressure, and Ruffini endings, responding to unchanging movements like stretching), and 2) *rapid adapting receptors* which are stimulated only at the onset and offset of a stimulus (Meissner corpuscles, sensitive to changing details, giving a perception of flutter, and Pacinian corpuscles, responding to changes in movement like vibrations).

At places where the tactile sensory capabilities are most acute (such as the fingertips), the spatial location of a point is detectable up to 0.15 mm, with a spatial resolution between two points of about one millimeter [13]. Textures made of a 0.06 μm high gratings are detectable, as well as 2 μm high single dots [13]. Vibrations of up to 1kHz are detectable, with highest sensitivity around 250Hz. The detection threshold globally decreases with increasing frequencies [13]. The frequency JNDs[1] at the fingertip has been estimated to different values among multiple studies [44] from 3% to 38%. The intensity JNDs at the fingertip decrease as intensity increases, and are roughly independent of frequency.

1.1.1.2 Proprioceptive receptors

Kinesthetic perception, often jointly used with the term proprioception, is involved in the perception of limbs' positions, movements and efforts. Proprioception is the result of the fusion of information generated by two kinds of receptors [13]:

- *receptors from muscles, joints and tendons.* The most important receptors for controlling the muscular system are the spindle fibers, sensitive to changes in the length of muscles, and the Golgi tendon organs, sensitive to stretch.
- *tactile receptors.* Tissues and ligaments surrounding the joint contain several mechanoreceptors such as Ruffini endings and Pacinian corpuscles, providing information such as the stretch of the skin.

Some studies suggest that the maximum bandwidth of kinesthetic perception is around 12Hz [45]. The position JNDs vary from 0.8°for the elbow to 2.5°for the fingers [45]. The perception of efforts is anisotropic. The typical associated JNDs are 5%−15% for contact forces, 10% for weight, 13% for torque and 22% for the stiffness of an object [46, 13].

1.1.1.3 Force control

During object manipulation, the maximum controllable force exerted through a finger is about 50 to 100 N, depending on whether shoulder muscles can be used or not [47]. However, typical forces in manipulation tasks are usually between 5 and 15 N, with a resolution of about 0.04 N [47]. When squeezing virtual objects, the perceptual resolution in terms of JNDs has been found to be about 7% for force and elastic stiffness, 12% for viscosity and 20% for mass [13]. In order to simulate rigid walls, a stiffness of about 25 N/mm is required, although 5 N/mm can already provide a good perception [47]. These thresholds, however, are the results of the haptic modality alone. They can be significantly altered when adding cues from other modalities [13].

1.1.2 Haptic Devices

When interacting with a VE with haptic feedback, a user receives tactual sensory information through his tactile and kinesthetic sensory systems. The interface in charge of displaying those haptic signals are the haptic devices. A particularity of haptic devices is that, in many cases and notably for kinesthetic interfaces, they also serve as input devices: the user controls and manipulates the device or part of it (such as a handle) through which positions, velocities and/or forces exerted by the user are sensed by the device, creating a two-way coupling between the user and the virtual object manipulated [13].

Haptic devices can be classified into tactile interfaces, involving tactile perception, and kinesthetic interfaces, involving kinesthetic perception. Since tactile perception is mostly cutaneous, tactile interfaces are usually used to simulate the direct touch and feel of objects contacting the skin [13]. Conversely, since kinesthetic perception is based on limb movement and net forces, kinesthetic interfaces have a handle or a grasping mechanical

---

[1] Just Noticeable Difference: the smallest detectable difference in a stimulus.

part, thus allowing users to feel the VE through a virtual tool called proxy. While haptic interfaces are not as common as visual or acoustic displays, numerous and diversified devices were developed.

Many factors define the perceptual quality and the capabilities of a haptic device: the number of input (sensing) and output (actuating) degrees of freedom, the maximum amplitude that can be displayed, the bandwidth, the motion range, the resolution, and the transparency [2], to name a few [47, 13, 48].

This section will briefly present some kinesthetic and tactile interfaces. A more comprehensive description can be found in [49, 50].

1.1.2.1 Tactile interfaces

Visell [31] characterizes tactile interfaces by the format in which energy is transmitted to the tactile receptors. The list is non-exhaustive:

- *Low frequency, low amplitude mechanical deformation*, where pieces are moved in order to render a relief for the exploration by touch. Examples include the tactile shape display [51] (Fig. 1.3a) for normal strains, and the STReSS interface [52] (Fig. 1.3b) for lateral strains.
- *Vibrotactile stimulation*, where the interface vibrates against the skin. Examples include the CyberTouch™ glove from Immersion (Fig. 1.3c) for finger and palm stimulation, and the EcoTile [31] (Fig. 1.3d) for foot-floor interaction.
- *Electrotactile stimulation*, where currents are used to stimulate the afferent nerves directly, bypassing the receptors.
- *Force feedback displays*, which by nature are designed for kinesthetic perception but inherently stimulate tactile receptors through contact transients (friction, vibrations, etc). Several examples are shown in section 1.1.2.2.
- *Thermal displays*, where heat is directed toward or away from the skin.

1.1.2.2 Kinesthetic interfaces

Kinesthetic interfaces are based on the kinesthetic part of the haptic system and thus measure and deliver positions and forces. We can differentiate passive interfaces containing only sensors and active interfaces delivering forces and movements to the user [53].

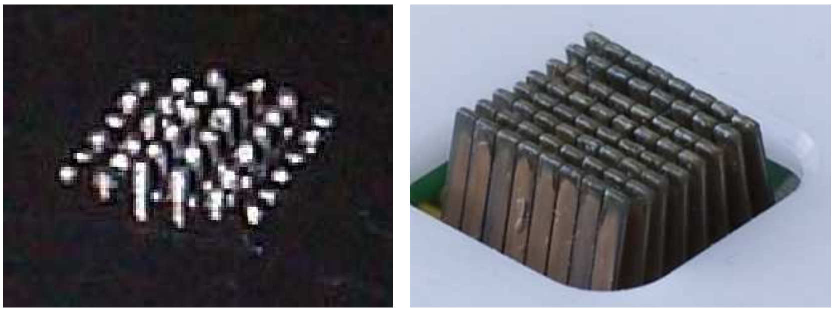
Passive interfaces regroup *isotonic* interfaces following the user's movements without constraining them and *isometric* interfaces which are, in contrary, immobile [54]. Existing passive interfaces include the classic mouse for position sensing, or the SpaceMouse™ from 3DConnexion for measuring forces and torques.

Active interfaces[3] can deliver forces using different kinds of actuation architecture, such as parallel or serial structures. In *serial active interfaces*, actuators are serially connected to each other from a static base to the manipulated part like a robotic arm. Notable examples are the Phantom® from SensAble (Fig. 1.4a) adapted to desktop use or the Haption Virtuose™ (Fig. 1.4b) allowing a wider workspace. In *parallel interfaces*,

---

[2] From a user perspective, a transparent system enables the haptic feedback of a VE without perceiving the mechanical dynamics of the device (such as inertia and friction)

[3] This class of devices is often referred in the literature as "haptic devices".



(a) Tactile shape display [51] **(b)** STReSS interface [52]

(c) CyberTouch™ glove **(d)** The EcoTile [31]

Figure 1.3 – Examples of tactile interfaces

actuators are directly connected to the manipulated part. Notable examples are cable-based devices such as SPIDAR interfaces [55] (Fig. 1.4c) or three-armed devices such like the Falcon® from Novint (Fig. 1.4d).

Other taxonomies classify active interfaces as ground-based (such as the Phantom®) or body-based (such as the Exoskeleton Force ArmMaster from EXOS, Inc., Fig. 1.4e) [13], or by the number of available degrees of freedom [49].

1.1.3 Haptic Rendering

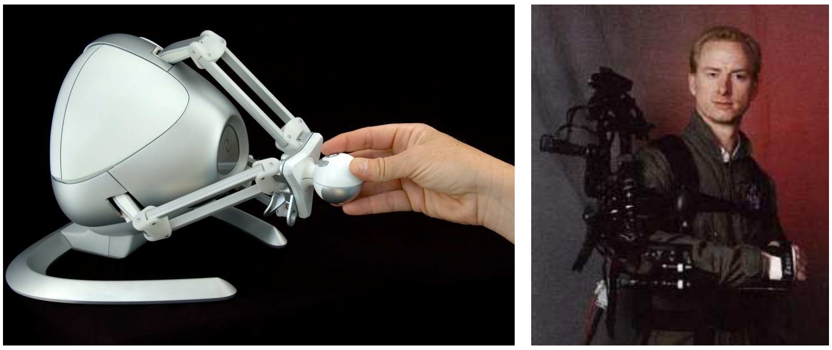
Haptic rendering is the component of machine haptics concerned with generating and rendering haptic stimuli to the human user [13].

A typical haptic loop consists of the following sequence of events [37]:

- Find the position of the proxy in the VE
- Use collision detection algorithms to detect proxy interaction with the VE
- Use data from collision detection and an interaction response algorithm to compute the interaction response signal
- Send the interaction response signal to the control algorithms, which apply them on the operator through the haptic device

The haptic rendering loop involves the user, the device and the simulation. The user controls the device, which senses a position/force that is sent to the simulation. In turn, the simulation generates the haptic stimuli, sent to the user through the device. Depending on whether there is a feedback mechanism between the device and the simulation, two different haptic rendering loops exist: closed-loop rendering, common for kinesthetic devices, and open-loop rendering, common for tactile devices.

(a) Phantom **(b)** Virtuose **(c)** Spidar [55]



(d) Falcon **(e)** ArmMaster

Figure 1.4 – Examples of kinesthetic interfaces

1.1.3.1 Closed-loop rendering

In closed-loop rendering, the output of the simulation is fed back to the input of the haptic device [56], as shown in Figure 1.5. This is typically the case of kinesthetic rendering, where a kinesthetic transducer displays a force to the user. Indeed, the force display induces a mechanical motion of the device, usually through its motors, that is eventually constrained by the user manipulating the device. Thus, there is a difference between the feedback configuration and the real device configuration, which is used to compute the new values sent to the simulation.

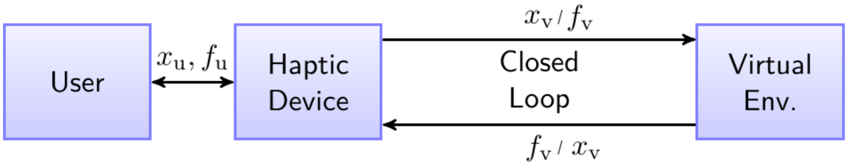


Figure 1.5 – Closed-loop rendering

Closed-loop rendering raises some specific issues. If an instability is introduced in the loop (due to rate or time-stepping issues, for instance), the closed nature of the rendering loop will make the errors propagate and will eventually be amplified until failure. Hence, in closed-loop rendering special care needs to be taken for the stabilization of the overall system. In addition, the loop has to run at an update rate appropriate to the type of haptic interaction that is simulated. An update rate of 1 kHz is considered to be a minimum in order to provide a good perceptive rendering, due to the frequency range of sensory receptors in the human haptic system. This rate seems to be a subjectively acceptable compromise permitting the representation of reasonably complex objects with reasonable stiffness. Higher rates can provide crisper contact and texture sensations for stiff objects, but only at the expense of reduced VE complexity or precision, or with the availability of more capable computers.

In general, closed-loop haptic rendering has to render two main features: the free motion corresponding to the unconstrained virtual object movement and the contact restraining the interface movement during collision. The efficient rendering of these features is highly dependent on the type of closed-loop control schemes and corresponding device. We can differentiate two broad classes of closed-loop control schemes: *admittance* and *impedance* [13]. The choice between these two main architectures raises some important implications in the design of the loop and the associated interface [56]:

- admittance control systems measure the force applied by the user and control the position and/or velocity of the haptic device. They can efficiently render contact surfaces by constraining the position regardless of the force applied but raises some difficulties to render transparent unconstrained free motion.

- impedance control systems detect the motion commanded by the user and control the forces applied by the haptic device. They can thus easily render free motions by not producing any force but cannot efficiently constrain the position on virtual contact surfaces.

The main issue raised by free motion and contact using admittance or impedance loops and devices is the instability caused by the rendering loop which can manifest as buzzing, oscillating or divergent behaviors, that might even be harmful for the user. We subsequently describe a mechanism called virtual coupling, designed to address this issue: it simulates virtual contacts based on a trade-off between a stable interaction and a convincing rendering of transparent free motion and stiff contact.

1.1.3.1.a Virtual coupling

The haptic interaction of a user with a VE can be described using Figure 1.5. x_{u} and f_{u} represent the forces and positions exchanged between the user and the haptic device, while x_{v}, and f_{v} represent the forces and positions exchanged between the haptic device and the VE. This configuration is called *direct rendering*, since the haptic device uses directly the data coming from the VE to display the force to the user. This approach, however, poses stability problems.

The simulation of a rigid contact can be modeled by using a contact model of stiffness K and damping B such that $f_{\mathrm{v}}(t) = -Kx_{\mathrm{v}}(t) - B\dot{x}_{\mathrm{v}}(t)$ [57]. This ideal spring-damper model is dissipative: an ideal spring is a lossless system (the energy accumulated by squeezing the spring is entirely removed when releasing it) and the damper is a dissipative system. However, the interactive simulation of such system implemented in discrete time and values is not dissipative. For instance, the spring will not increase smoothly, but will be repeatedly “held” at constant values of time-step T in a staircase fashion, and the energy accumulated during squeezing will not be entirely removed during release. Thus, the virtual spring does not behave as a lossless system but generates energy. Likewise, a discretized damper is capable of producing energy. This discrete sampled-data system constitutes one important reason of the instability of haptic rendering of virtual contact [57].

Colgate *et al.* studied the stability [57] of the above system for the simulation of a rigid contact modeled by such a discretized spring-damper. They introduced the idea of a sufficient condition for stability in the idea of *passivity*, describing the fact that the system should not generate energy. Then, they proposed a general approach guaranteeing passivity by introducing a *virtual coupling* between the haptic device and the interactive simulation [58], such that output forces and torques are not rendered directly to the user.

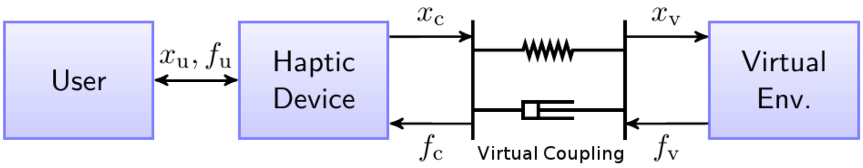


Figure 1.6 – Virtual coupling mechanism: a visco-elastic link in the form of a spring and a damper is introduced between the device and the VE.

The virtual coupling is a multidimensional viscoelastic link (spring and damper) between the haptic device and the simulation, as illustrated in Figure 1.6. Thus, the position of the virtual object in the simulation is distinct from the position imposed by the haptic device, but both are connected by a viscoelastic link. The spring tries to align the simulation position of the virtual object to the position of the haptic device. The damper tries to enforce equal velocities. This link exists for position and orientation in the case of 6DoF simulations [58].

Virtual coupling guarantees the stability of the discrete-time sampled system by limiting the maximum impedance exhibited by the haptic device as long as the simulation is discrete-time passive. Thus, even if the simulated object is constrained by a rigid contact (with high values of stiffness and damping) from the simulation, the virtual coupling will naturally limit those values to achieve a stable haptic rendering. The use of virtual coupling shifts the stability problem of the haptic rendering of a VE to the passivity of the simulation alone. Colgate *et al.* pointed out that this passivity of the simulation can be easily achieved by considering methods with implicit integration schemes [58].

1.1.3.2 Open-loop rendering

In open-loop rendering, the display of the force stimuli has no incidence on the data sent to the simulation [56], as shown in Figure 1.7. This is typically the case of tactile rendering, where a tactile interface displays force transients or any other signal to the user by, for example, vibrating, without incidence in the position data sent to the simulation. The loop is inherently more stable than a closed-loop rendering approach [56].

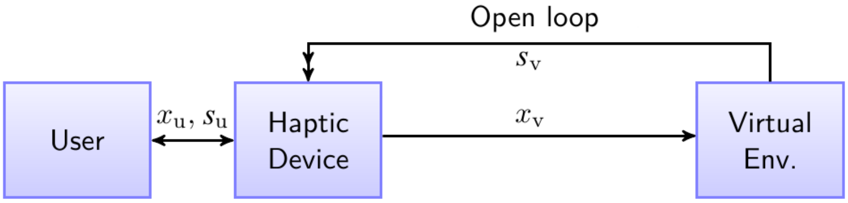


Figure 1.7 – Open-loop rendering: the output signal s_v have no incidence on input data.

In the case of a kinesthetic device, open-loop can be achieved by momentarily ignoring the measured position of the device during the duration of the event [59]. It allows to present an impact transient pattern through an adequate kinesthetic device. This combined approach has been extensively used to enhance the perception of contacts by producing a contact pulse force pattern superimposed to the regular closed-loop rendering [60, 61, 59].

1.1.3.3 Simulation loops

Haptic rendering architectures usually combine haptic feedback with visual feedback. The classic architecture relies most of the time on multiple sensory rendering loops communicating with a physical simulation loop in a synchronous or asynchronous way [62]:

- The physical simulation loop is usually the main loop describing the movements and interactions between the virtual objects. This loop receives position and force inputs from the manipulation devices and updates the simulation accordingly. Then, multiple output information describing the VE are provided to each modal rendering loop. The physical simulation can be performed using multiple loops to separate the processing of collision detection and dynamic movement resolution, for instance.

- The kinesthetic rendering loop (closed-loop rendering) controls the force-feedback restitution. An update rate of 1 kHz is generally considered a minimum to achieve a perceptively good and stable rendering. This loop is often included in the physical simulation loop to easily achieve the bilateral processing of force/position input and output with the physical simulation.

- The tactile rendering loop (open-loop rendering) receives information from the physical loop to provide tactile display. The update rate of this loop depends on the specific tactile features being rendered (deformation, vibrations, temperature, etc.) which can vary from several Hertz to several kiloHertz.

- The visual rendering loop generates images from the simulation. The minimum frame rate is usually considered to be around 20 Hz. However, 60 Hz are required to provide a smooth rendering. Stereoscopic viewing requires a two-times higher frame rate, and additional information about the position of the viewer is needed to adapt the view accordingly.

One important aspect of this architecture is the underlying synchronous or asynchronous communication layer and particularly the latencies generated which must remain as low as possible to preserve the multimodal experience.

These human and machine haptics fundamentals set the perceptual, mechanical and architectural building blocks, upon which different haptic rendering models will be developed. These models depend on the type of object that need to be simulated, and the sensory channel that will be used to render the feedback, as we will see in the following section.

1.2 Models for Physically Based Haptic Interaction

In a haptic VR application a user usually holds the active end of a haptic device, represented in the VE as a tool (proxy), in order to interact with the objects of the VE. In order to simulate the interactions and the different phenomena that arises during the simulation of the VE, we need a model to describe the dynamic behaviors between the proxy and its interacting virtual objects. This model should be fine enough to capture the real world interaction experiences in a form of a force/torque or/and tactile feedback. This model is usually transversal to the simulation steps of the haptic rendering component [63], which are:

- Finding the contact point(s), which involves collision detection algorithms of variable precision and complexity depending on the underlying model
- Generation of contact forces, depending on the different input data supported by the model (stiffness, damping, friction, surface texture, normal, etc)
- Dynamic simulation of the VE, updating the VE to account for the effect of the interaction on the simulation side

Hence, the underlying model plays a fundamental role in the quality and the performance of the resulting haptic feedback. In addition, computational rates must be high (at least 1kHz or higher for rigid bodies), otherwise hard surfaces in the VE might feel soft and system instabilities might arise.

Many different models exist, depending on the requirements of the simulation. Algorithms range from geometry-based collision models, providing a simple but low precision approach, to physically based algorithms allowing rich and precise interactions at the cost of a complex and computationnaly expensive simulation.

Most models are specific to one particular type of matter, namely rigid, deformable or fluid states. The main reason is that by constraining the state of the environment, one can make several assumptions that can dramatically simplify computations. For example, both rigid and deformable bodies are solid objects and therefore follow the same physical behavior. However, rigid objects such as stone or metal exhibit unnoticeable deformations under common loads. Hence, they can be more efficiently approximated as an infinitely stiff body following its own dynamics. Fluids, conversely, exhibit high deformation and topology changes compared to solids. Hence, other data structures and computational techniques will be more appropriate for their efficient simulation.

In this section, we survey the different models for the haptic interaction with rigid, deformable and fluid matter. With haptic we refer to both kinesthetic and tactile feedback. However, when addressing tactile feedback we restrict the scope to *vibrotactile* feedback, due to the common availability of “generic” devices for this specific modality. Other components of tactile feedback require very specific and often expensive devices (such as mechanical moving parts, electrotactile displays and thermal displays), thus currently limiting their impact in VR. Nevertheless, there is a wide range of phenomena that can be rendered through vibrotactile feedback.

Since most haptic devices available nowadays allow 6DoF input and 3DoF or 6DoF output, and considering the significant computational power available nowadays, we do not survey the models that focus on single point (3DoF) rendering. In addition, since our research context sets physically based VE as a fundamental condition, we focus on physically based models.

1.2.1 Rigid Bodies

Rigid bodies are approximated as infinitely stiff objects. As a consequence, they do not undergo any deformation, making the computations of their dynamics simple compared to other states. Consequently, rigid bodies can model real matter that exhibits negligible deformations, which represents many, if not most, of the objects we manipulate and interact with in a daily basis. Different approaches are used for kinesthetic and vibrotactile feedback.

In the case of kinesthetic rendering, the motion of the entire body is taken into account. Rigid bodies follow rigid body dynamics, the laws and equations that govern the behavior of rigid bodies, described in Appendix A. Since there are no internal forces (infinitely stiff body), only 6 degrees of freedom need to be computed at each time step (3 in translation, 3 in rotation) at the center of mass of the body, making the computation fairly straightforward. The challenge for kinesthetic rendering lies in the high update rate requirements, which are particularly important for rigid bodies if a stiff contact needs to be rendered. Accurate and fast collision detection needs to be performed to avoid perceivable interpenetrations, and adequate and realistic forces need to be generated. Kinesthetic interfaces are used to display the forces, with closed-loop control.

In the case of vibrotactile rendering, the rigid body is still an infinitely stiff object from a macroscopic modeling and deformation point of view. However, vibrotactile rendering of rigid bodies considers the vibrations (often invisible) generated at impact or during frictional contacts, transmitting many contact and material cues to the user. Deformations are not explicitly modeled, but instead physically based oscillation equations are used to model vibrations. Vibrotactile interfaces are used to display the signals, with open-loop control.

1.2.1.1 Kinesthetic rendering

To enable the 6DoF physically based kinesthetic rendering of rigid body interaction, two high level approaches have been proposed, which correspond to the way the rigid bodies are represented: polygon-based models for triangle intersection queries and voxel-based models for point-voxel intersection queries [64].

1.2.1.1.a Polygonal models

Gregory *et al.* [65] provide the first polygon-based 6DoF haptic feedback algorithm by using the Lin-Canny closest points algorithm [66] as the basis for the collision detection. The algorithm determines the closest points by tracking the closest features (vertex, edge or face) on the surfaces of the two interacting objects. Since the tracking algorithm is less efficient when the two objects are interpenetrating, the authors assume a contact has occurred when the distance between the features is less than a fixed distance δ. A penalty force is computed using the resulting penetration depth. Penalty forces, widely used in haptic rendering approaches due to their simplicity and inexpensive cost, are based on the amount of object interpenetration [67]. If δ is the penetration depth and $\mathbf{v}$ is the relative velocity along the contact normal, the penalty force is usually formulated as $\mathbf{f} = -k\delta - b\mathbf{v}$, where k and b are fixed arbitrary stiffness and damping values.

A similar approach was adopted by Johnson and Willemsen [68], where they create a large buffer zone around the virtual objects. The algorithm uses the buffer zone to prevent collision instead of using the slower approach of computing the actual penetration. These distances are computed using Spatialized Normal Cone Hierarchies [69], and penalty forces are generated.

Kim *et al.* [70] developed a technique where the resulting haptic force is computed using the real penetration depth between the proxy and the interacting object, leading to smoother and more stable forces. They employ an iterative local optimization method that finds a locally optimal solution by walking along the surface of the Minkowski difference. A contact clustering is also proposed to overcome the issue of having multiple redundant contact points, by grouping contact points under a threshold. This scenario can arise in approaches previously described which work only for convex pieces. Concave pieces need to be decomposed, potentially generating a high number of convex pieces, which in turn can potentially cause many contact points.

A different approach consists in using constraint-based methods for stiff and non-interpenetrating collisions. Ortega *et al.* [71] extend the god-object [72], a 3DoF constraint-based technique, to achieve 6DoF haptic coupling and rendering. The unconstrained acceleration of the proxy (called god-object) is computed considering the distance between the proxy and the haptic device. This acceleration is then updated by ensuring the proxy is constrained to the surface of the contacting object. A new proxy configuration is computed, and continuous collision detection is used [73] with the old and new configurations to find the next contact points and the final proxy configuration. The force transmitted to the user is a penalty force based on the constrained acceleration. The authors argued that the coupling they introduced provides a better force rendering than the classic viscoelastic virtual coupling [71], as shown in Figure 1.8.

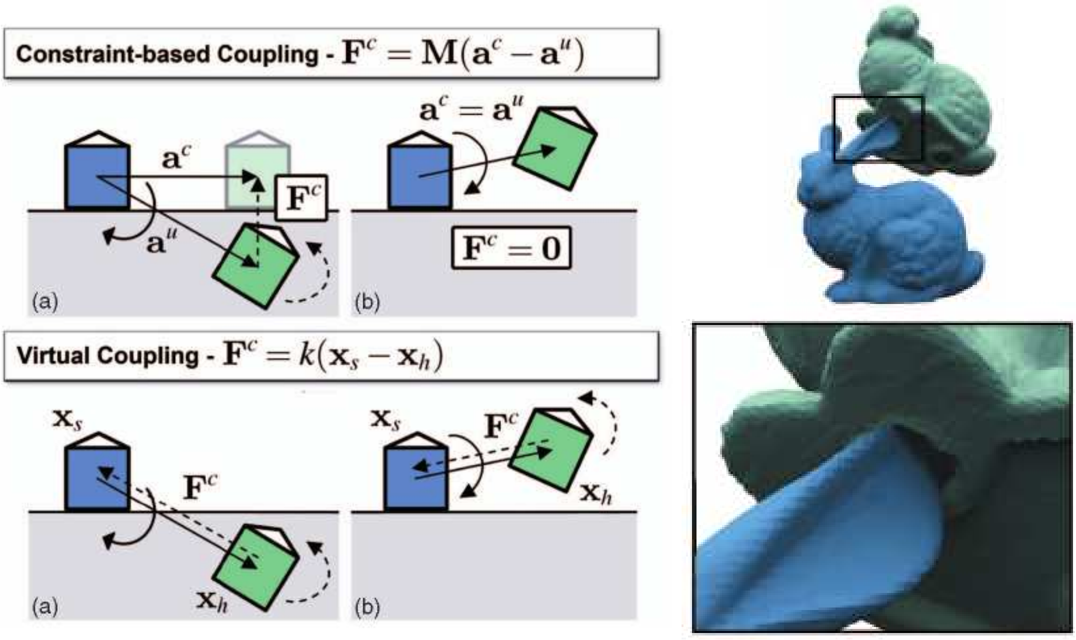


Figure 1.8 – 6DoF God-object technique. Left: the coupling mechanism of the god-object removes force the artifacts of virtual coupling: (a) in a penetration case, a normal force is generated instead of an incorrect tangential force, (b) in a detachment case, no force is generated, instead of the usual spring force of virtual coupling. Right: the approach allows high precision contacts with haptic feedback [71].

Several approaches use multiple levels of detail or multirate simulations in order to improve haptic rendering. A sensation-preserving simplification algorithm [74] trades accuracy for speed through multiple levels of detail while preserving the interaction forces by selecting contact resolutions adaptively. This technique is further improved [75] with an implicit integration scheme, a linearized contact model and a multirate approach for the decoupling of the proxy simulation and the collision detection and response. A fast contact point tracking algorithm based on spatialized hierarchies is used by Johnson *et al.* in [76] for approximating the collision detection. It is combined with a slower algorithm for exact computations, and allows the use of moderately sized moving objects at haptic rates.

Other haptic rendering techniques focused on adding forces from small scale texture details on top of common low-frequency interpenetration forces during the sliding part of the contact. These techniques allow the display of high-frequency geometric texture detail through new representations for the detailed geometry. Relying on high resolution collision detection and high density sampling would otherwise be prohibitively expensive. Minsky [77] conveyed texture information by displaying a texture-induced force proportional to the gradient of a 2D height field stored in a texture map. A stochastic approach is used by Siira and Pai [78], where texture forces are computed according to a Gaussian distribution. These techniques only allow point-based and one texture map (hence, only one textured object). Otaduy *et al.* [79] address these problems by introducing force and torque computation for interacting textured objects. They state roughness perception is tightly coupled with the oscillation of penetration depth, and therefore use the variation (gradient) of the penetration depth instead of the penetration depth itself for computing penalty forces.

1.2.1.1.b Voxel-based models

Dividing the VE into a grid of voxels is an alternative to using polygons for distance and intersection queries between two objects. Voxel-based hierarchical spatial subdivision is inherently faster for collision and distance tests thanks to optimized tree traversals, but at the cost of a loss in boundary accuracy. The loss depends on the voxel resolution, and can therefore be arbitrarily set.

McNeely *et al.* [80] proposed the first voxel-based 6DoF haptic rendering technique, allowing the haptic interaction with complex rigid bodies as in Figure 1.10. The proxy and the VE are converted into a 3-level voxel hierarchy based on a 512-tree. The collection of voxels subdividing the VE is called Voxmap. The centers of the voxels of the proxy surface are sampled into a set of points called PointShell. The PointShell also holds the inwards normals to the proxy surface at each sampled point. The Voxmap and Pointshell are illustrated in Figure 1.9 (left). In order to compute the interaction between the Voxmap and the PointShell, the voxel tree is used to efficiently find a contact with the points of the PointShell. Then, for each contact, a force vector is computed as a penalty force. The penetration depth used for the penalty force is the distance between the contacting point of the PointShell and a plane passing through the center of the contacting voxel and orthogonal to the corresponding PointShell normal, as shown in Figure 1.9 (right). All the forces are then applied to the proxy and integrated, obtaining a new proxy configuration which is transfered to the haptic device through virtual coupling [58] in order to display a feedback to the user.

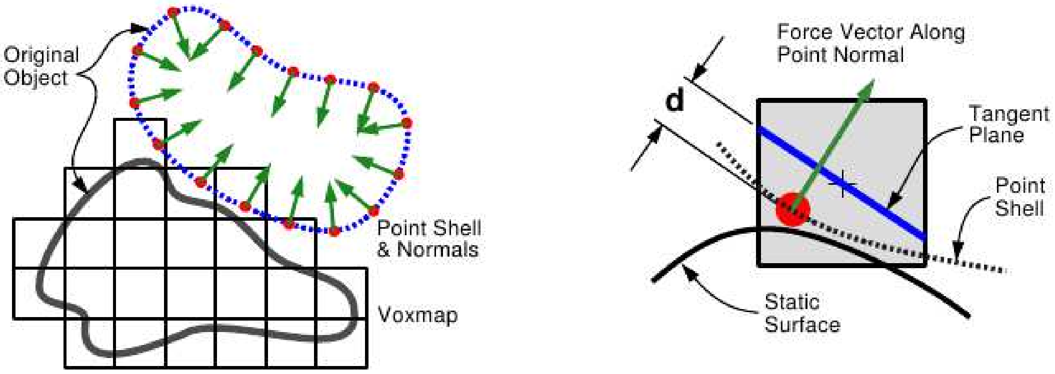


Figure 1.9 – Voxel-based rendering. Left: the Voxmap and PointShell discretizations. Right: penetration depth computation [80].

Several improvements where developed on top of the original approach. Renz *et al.* [81]

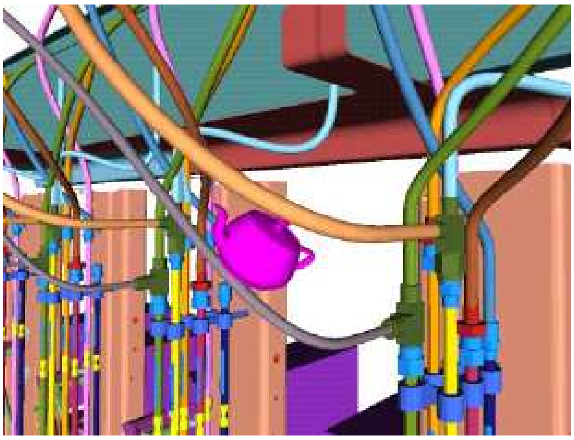


Figure 1.10 – Example of voxel-based rendering allowing the interaction with complex VE at haptic rates [80].

projected the voxel centers forming the PointShell onto the surface of the triangulated objects, to obtain a smoother surface representation. Wan *et al.* [82] improve the stability of the system in the case of a static VE by computing the dynamics of the proxy through a quasi-static approximation. The authors solve a system of spring forces, including the contact penalty forces and the virtual coupling spring, for static equilibrium and obtain a new proxy configuration. In [83], a voxel-based distance field is used for an advance warning of potential contact by extending the voxelization beyond the objects surface and estimating a distance-to-surface value in number of voxels. Geometrical awareness is also included, by labeling each voxel as vertex, edge or surface if they contain these features. Then, for each feature a voxel-based distance field is computed accounting for the distance in voxels to the nearest feature. These additional distance fields enable the culling of point-vertex intersections at runtime during tree traversal, where only vertex-surface and edge-edge contacts are interesting. Further tree traversal optimization is done by exploiting the time coherence of the status of the points in the PointShell, which are likely to remain in contact after a contact in the previous frame. Ruffaldi *et al.* [84] take advantage of the simpler sphere-sphere intersection test compared to cube-cube by using a sphere space subdivision instead of voxels.

1.2.1.2 Vibrotactile rendering

Several techniques have proposed high-frequency open-loop force transients for the haptic rendering of rigid body contact. These transients were either superimposed to the kinesthetic rendering approaches previously described or rendered directly through a vibrotactile transducer. They respond to collisions between the proxy and the VE, and are therefore generated at the beginning of the contact. We can distinguish approaches generating braking forces, used to improve the perception of highly stiff materials, and contact perception approaches transmitting vibrotactile cues of the type of rigid material being simulated.

1.2.1.2.a Braking forces

Transient braking forces have been used at the moment of impact to enhance stiffness perception. Salcudean *et al.* [60] compute the force transient as the one required to bring the proxy velocity to zero in an explicit Eulerian integration scheme. In voxel-based rendering, McNeely *et al.* [80] also generate a pre-braking force using the PointShell velocity multiplied by a “braking viscosity” coefficient and projected onto the surface normal, in order to avoid significant interpenetration and hence unstable intersection resolutions.

1.2.1.2.b Contact perception

Other transient forces have been used at the moment of impact to improve the perception of contact with rigid bodies of different materials. Okamura *et al.* [85] gathered high resolution real acceleration information by tapping on different materials. This data was fit into a decaying sinusoidal signal, with the amplitude depending on the proxy velocity. Each captured material is represented by a different vibration signature with specific frequency and decay rates. This vibrotactile signal is then rendered in open-loop fashion through the haptic device at the moment of impact, successfully conveying perceptual information about the material stiffness.

However, transducers do not exactly reproduce what is recorded through other devices. Hence, Okamura’s technique was later improved [61] to compensate for device dynamics and rendering bandwidth by applying scaling factors found through human perceptual experiments for a given transducer, as illustrated in Figure 1.11. This led to the rendering of three realistic signals corresponding to rubber, wood and aluminum, but had the main drawback of requiring a perceptual tunning step for each transducer. Kuchenbecker *et al.* [59] addressed this issue by adapting the signal to the dynamic response of the device using an inverted system model of the display. This enabled the rendering of realistic impacts from recorded force patterns. Further studies by Fiene *et al.* improved this technique by considering the grip force applied to the device [86].

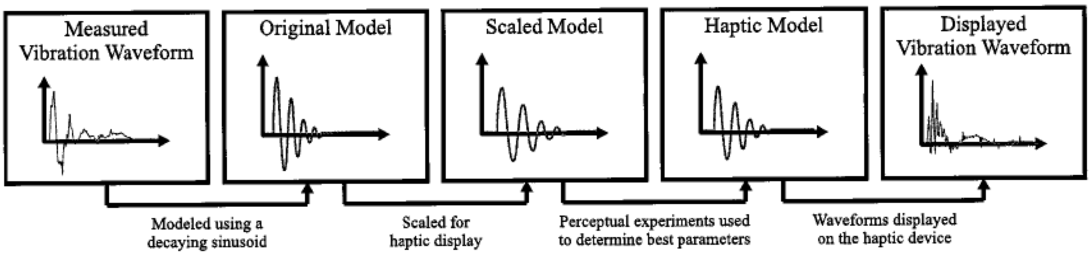


Figure 1.11 – Vibrotactile rendering of rigid contacts: different steps from vibration measurement to vibration synthesis taking into account device and perceptual parameters [61].

Since these techniques were originally designed for an impact along one dimension, Sreng *et al.* [2] proposed an approach for 6DoF open-loop contact transients, using force and torque contact data and a constant oscillation frequency. The resulting signal is superimposed to 6DoF kinesthetic feedback in a virtual assembly task. The approach is extended by the same authors [87] to provide cues about the distance to the contact point when interacting with the VE through a rigid beam, as illustrated in Figure 1.12. Amplitude and frequencies of the oscillatory output signal depend on the virtual distance from the device handle to the contact point in the VE.

1.2.2 Deformable Bodies

In the previous section, we surveyed haptic rendering techniques for rigid body proxies and VE. Dealing with deformable bodies, however, is more challenging than dealing with rigid bodies, for simulation and computational reasons. Deformable objects follow the physical phenomena of continuum mechanics, which are mathematically and hence computationally

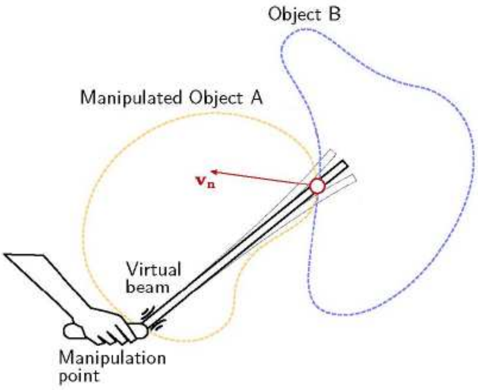


Figure 1.12 – Spatialized haptic rendering: the virtual rigid beam vibrations provides cues about the contact distance [2].

much more complex than rigid body dynamics. In addition, due to the inherently dynamic shape of deformable bodies, collision detection becomes more expensive, especially for contact detection. Higher simulation complexity means higher computational load, and hence achieving the 1000Hz update rate common in haptic simulation is far from trivial.

There is again an important distinction when rendering through the kinesthetic and the vibrotactile modalities. The use of expensive deformable models in interactive applications is justified when deformations are visible, and therefore for high amplitude and low frequency deformations. High amplitude and low frequency signals are well suited for kinesthetic rendering. However, their display through the vibrotactile modality poses many additional challenges and limitations. Vibrotactile transducers are taken to the limits of what they can display in terms of amplitude and frequency, and vibrotactile receptors in the skin loose sensibility under 10Hz [13]. Therefore, while kinesthetic feedback will be used for objects following the laws of continuum mechanics with visible deformations, vibrotactile feedback will be used for other types of deformable materials, such as granular media, where energy transients are released during configuration changes.

1.2.2.1 Kinesthetic rendering

Thankfully, the computer graphics community has been very active in the area of deformable body simulation, and the idea of haptic interaction in the medical medical field for surgical simulation applications has motivated research for the addition of haptic feedback [64, 88]. There are many different techniques build upon different models, such as mass spring systems and the Finite Element Method.

The details of continuum mechanics, the laws that govern the behavior of deformable bodies, together with the different discretization models generally used in real-time physical simulations are described in Appendix A. Here, we survey the existing kinesthetic rendering techniques for deformable bodies, following their underlying discretization models: mass-spring systems, the Finite Element Method (FEM) and mesh-free systems.

1.2.2.1.a Mass-spring systems

Mass-spring systems are among the simplest approaches for the simulation of deformable bodies, with a set of point masses linked together through springs and dampers. More details are given in Appendix A.

Early approaches adopted mass-spring system models due to their low computational costs. The first reported technique, from Swarup and Salisbury [89], goes back to 1995. Theu used a linear mass-spring-damper system, with collision detection achieved by testing each node against a spherical region corresponding to the proxy's action radius. A penalty force was applied on the nodes and the proxy. In order to achieve real-time speeds for the available hardware, only the local nodes in contact with the proxy were simulated. Corso *et al.* [90] developed a technique using the Medial Axis Transform for the interaction with a deformable parametric surface. A medial surface B-Spline represented an implicit volume defined by a set of connected spheres with variable radius. It used a mass-spring system for the deformable simulation which affected the control points of the medial surface and the sphere radii, and penalty-forces for haptic rendering. Weiss and Okamura [91] allowed the cutting of a 2D mass-spring-damper system. Two models were used, one for the simulation of the mesh, and the other for the simulation of the cutting. The system was coupled to 2DOF haptic scissors for cutting in a medical context.

Some mass-spring system haptic rendering approaches have incorporated the simulation of non-linear deformations. Cavusoglu and Tendick [92] used a local linear approximation of a full non-linear mass-spring-damper system for a multi-rate simulation. They argued their local approach could provide accurate results since the linearization depended mainly on neighbor nodes. d'Aulignac *et al.* [93] measured the deformation of the surface of the thigh in order to extract its non-linear stress-strain curves. Then, they modeled the thigh as a two-layer mass-spring system, with linear springs at the surface and non-linear springs underneath orthogonal to the surface, as illustrated in Figure 1.13. The parameters of the non-linear layer were estimated by fitting the measured data using a least-squares minimization method. An implicit integration with a conjugate gradients solver is used with very coarse models. Force feedback is generated through penalty forces.

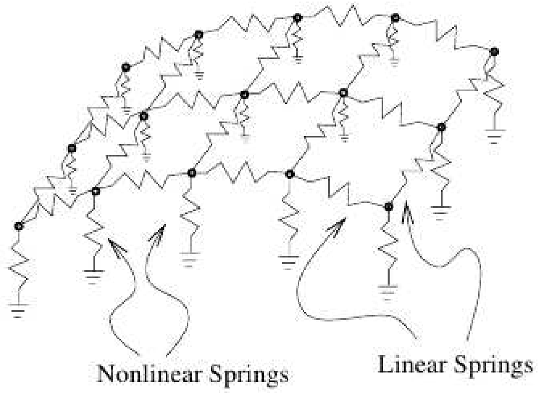


Figure 1.13 – Non-linear deformations using mass-spring systems: two-layer configuration using linear springs at the surface and non-linear springs underneath [93].

1.2.2.1.b Finite Element Methods

The Finite Element Method (FEM) treats matter more as a continuum, with objects represented as a set of contiguous elements (such as tetrahedra) that deform under applied loads, as if the vertices of each element were connected by three-dimensional hyper-springs. More details are given in Appendix A.

Early approaches used a linear elastic deformation model. Although this approach is only valid for small deformations and strains, it has the advantage of leading to a linear matrix system which is easier and faster to solve. This linear elastic model is often combined with linear FEM, which uses a constant stiffness matrix throughout the simulation, evaluated at the equilibrium state as a preprocessing stage. It avoids the recomputation of the matrix at each time step but further limits the simulation to small deformations. In [94], a linear elastic model with linear FEM is used, with an additional step called condensation that reduces the stiffness matrix of the volumetric mesh to a system with the same complexity as a surface mesh, but still simulating a volumetric behavior. The system is solved directly using the inverted stiffness matrix, computed in a preprocessing step regardless of the eventual numerical errors introduced due to the inversion. Cotin *et al.* [4] use a displacement-driven approach, where bilateral constraints are applied at the contact point instead of forces. The constraints are solved using Lagrange multipliers. Proxy forces and mesh deformations are computed in real-time using a linear elastic model and precomputed force and displacement tensors for each node. The collision detection is based on a cell space partitioning and a static sorting of the nodes per cell with a hash table for fast access, and assumes the deformations are small and hence the nodes stay within the neighboring cell region. In a similar approach, Popescu *et al.* [95] propose the small area touch paradigm, where only the nodes in close proximity to the proxy are updated using a small local stiffness matrix. Collision detection is done on a fine version of the mesh, where the displaced positions on the fine mesh due to proxy contact are interpolated back to the coarse deformable mesh.

Other approaches relying on force and displacement precomputation [96, 97] used the more complex viscoelastic and non-linear models. They could still achieve stable and fast haptic rendering by interpolating the precomputed data at haptic rates. James and Pai [98] presented an approach for the interaction with deformable media with elastostatic contact simulation, using a contact resolution based on a capacitance matrix. The capacitance matrix relates contact displacements to force feedback response. The method relies on the precomputation of the deformation basis using Green's functions. All these techniques, however, are limited to point-based interaction due to the single contact point limitation of precomputations.

Barbic and James [99] tackle collision detection and contact resolution problems for 6DoF haptic interaction with highly detailed deformable VE, as shown in Figure 1.15. They design a voxel-based collision detection algorithm with a multiresolution pointshell for a progressive computation of contact forces. The additional challenge compared to voxel-based rigid body approaches is the computation of the signed distance between a pointhsell point and the voxelised object, knowing that recomputing the distance field at each deformed state (i.e. at each timestep) would be far too expensive. This is solved by using a coarse set of pointshell points on the voxelised object that are deformed with the object, called proxies. At each distance query, a k-nearset neighbor search finds the closest proxies to the query point, and a local first-order deformation is computed for each selected proxy. The signed distances can be computed using the inversed deformation transform and the distance field precomputed at rest, and the final approximated distance is computed by weighted interpolation. This process is illustrated in Figure 1.14. Penalty forces and a customized virtual coupling mechanism are used for force feedback. A non-linear reduced FEM model is used for computation of deformations, but their approach can be applied to other models as well.

The computation of forces through penalty-based methods, although fast and efficient, present many drawbacks such as interpenetration, lack of physical accuracy, dependency on the penalty stiffness parameter, and limited integration with friction forces [100]. To address these issues, constraint-based methods were introduced, with improved contact handling at higher computational costs. Duriez *et al.* [5] formulate contact constraints

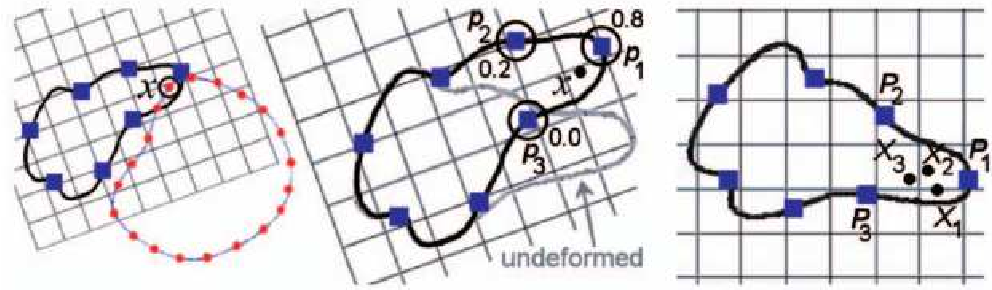


Figure 1.14 – Voxel-based rendering of deformable bodies: the different steps of the deformed distance field approximation. Left: the proxies (squares) and the query pointshell point at x. Middle: the k-nearest neighbors and their weights. Right: configuration of x in the undeformed state computed by weighted interpolation [99].

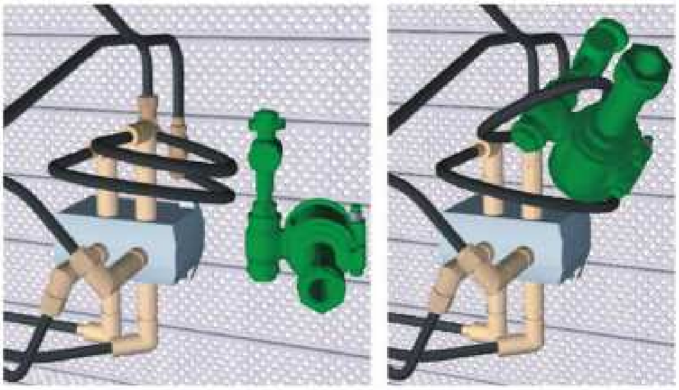


Figure 1.15 – Example of voxel-based rendering of highly detailed deformable bodies [99].

as a Linear Complementarity Problem (LCP), taking into account the compliance of deformable bodies through linear FEM. Friction forces are also formulated and integrated into the LCP. The system is solved using a Gauss-Seidel-like iterative algorithm. Thanks to the ability to solve contact resolutions between two deformable bodies, the approach allows the interaction through a deformable proxy, as shown in Figure 1.16. However, the LCP algorithm requires an inversion of the stiffness matrix. Hence, linear FEM is used to enable its precomputation and inversion, although restricting the approach to small deformations. In order to estimate the inverse of the stiffness matrix for non-linear FEM, Saupin *et al.* [101] also precompute the stiffness matrix, but relies on corotational FEM and the "warping" of the precomputed matrix. The "warping" takes advantage of the corotational approach [102], and uses a local evaluation of the rotations to estimate the contact compliance at each time step. They use a multirate approach with a visual thread, solving the LCP, and a haptic thread, solving the LCP with additional constraints coming from a god-object [72] haptic coupling. The solution of the LCP in the visual thread is used in the haptic thread as a good guess for the fast convergence of the haptic LCP resolution. With corotated FEM, haptic and visual feedback from large deformations remain accurate, which is particularly attractive in medical simulators (Figure 1.17).

1.2.2.1.c Mesh-free models

The main difference between mesh-free (or meshless) models and other deformable models is the lack of connectivity information. Since meshless models do not store this information, it has to be reconstructed at each time step. This has the advantage of allowing the

Figure 1.16 – Linear deformations using an LCP formulation: snap-in (top) and snap-out (bottom) tasks on deformable pipes with deformable proxies and friction feedback [5].

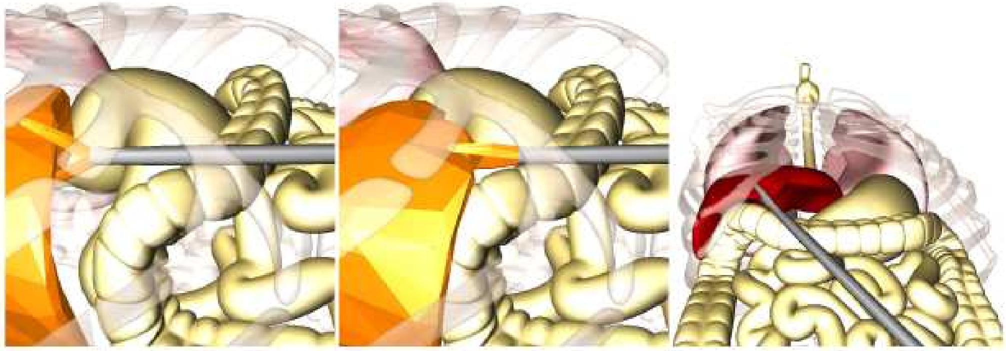


Figure 1.17 – Non-linear deformations using an LCP formulation: applications in a surgical simulator [101].

simulation of topology-changing objects without additional cost. Meshless models allow simulations such as cutting to be seamlessly performed where polygon-based models require expensive remeshing steps. They also greatly reduce the complexity of surface and volumetric object generation.

Physically based mesh-free or meshless models have attracted the attention of graphics researchers in the last decade. However, only one approach takes partially advantage of their flexibility in the haptic realm. In order to enable the cutting of a polygonal object deformed through FEM, De *et al.* [103] introduced the Finite Sphere Method. When the proxy contacts the polygonal object, a set of points is created locally around the tip of the proxy, as illustrated in Figure 1.18 . A sphere with a finite radius is created around each point. The governing partial differential equations of linear elasticity are applied at the points, and the spheres define an influence zone where Moving Least Squares are used for the interpolation. A force extrapolation technique is then used to obtain real time performance.

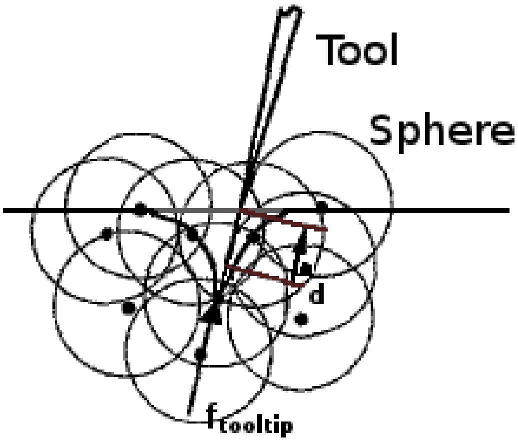


Figure 1.18 – Meshless rendering: placement of the nodes at the proxy tip (tooltip) contacting the deformed surface. d is the displacement, and $f_{tooltip}$ is the resulting reaction force [103].

1.2.2.2 Vibrotactile rendering

There is a small body of literature addressing the vibrotactile rendering of deformable media exhibiting low frequency deformations [104]. This includes the rendering of granular media, as well as recent work on simulating material compliance through vibrotactile stimulation.

1.2.2.2.a Granular materials

Granular materials, such as sand or gravel, are large conglomerations of discrete macroscopic particles. From a macroscopic point of view, we consider granular materials as deformable media since they exhibit viscoelastic behaviors due to recoverable local deformations around the contact point, as well as plastic behavior from the collapse of brittle structures or granular force chains, resulting in unrecoverable deformations [31].

Visell *et al.* [105] user a simple yet physically motivated approach for the vibrotactile rendering of the interaction with granular materials. The approach is based on a minimal fracture mechanics model [50], using a Mohr-Coulomb yield criterion to determine slip events within the structure, leading to an energy loss that can be synthesized as discrete transients. High frequency vibrations are generated when interacting with the material. The approach is illustrated in Figure 1.19.

This model is used in conjunction with the EcoTile [16, 32], a composite tile equipped with a vibrotactile transducer allowing the paving of an actuated floor surface. Walking on the tiles gives the illusion of stepping on natural ground materials such as sand, snow [105] and ice [32], as shown in Figure 1.20.

1.2.2.2.b Compliance rendering

Recent work [104] has addressed the challenge of simulating material compliance through vibrotactile stimulation, with very promising results. The study focused on ground material and thus foot-floor interaction. The authors showed that ground compliance perception could be altered through vibrotactile feedback, even for amplitudes close to the human perception thresholds. Using a modified version of the EcoTile [16], subjects walked on the vibrotactile tile indicating the compliance they felt in subjective magnitude or pairwise

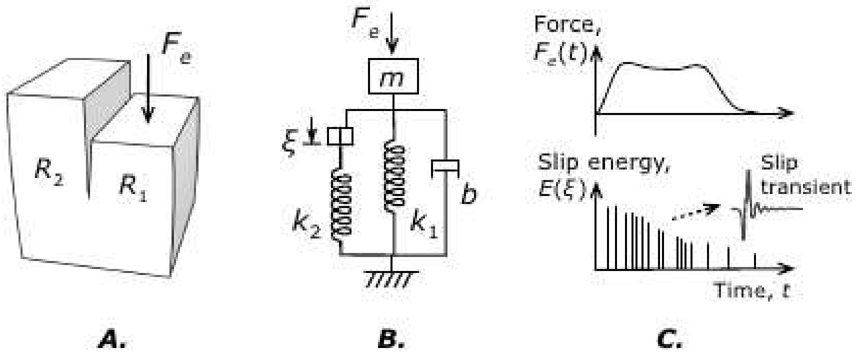


Figure 1.19 – Fracture mechanics approach for the vibrotactile rendering of granular materials. Left: a load F_e generates shear sliding fracture in a visco-elasto-plastic body. Middle: the mechanical analog to the fracture, generating slip events ξ. Right: slip events are rendered as transient impulses [32].

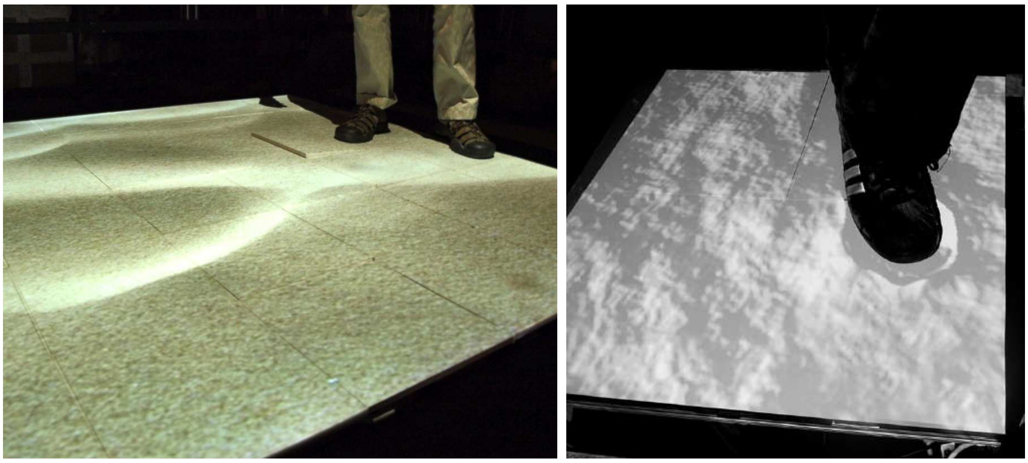


Figure 1.20 – Examples of vibrotactile rendering of granular materials. Left: the sand floor. Right: compacted snow with the user footprint [105]

comparison. The perceived compliance of the floor tile was shown to increase with the intensity of the vibration feedback signal, while depending to a lesser extent on temporal parameters. The effect persisted even for floor tiles of different real stiffness.

The different approaches for haptic interaction with deformable bodies presented in this section are highly dependent on the underlying physical model. This is particularly the case for kinesthetic rendering models. Mass-spring system are among the simplest and thus fastest approaches. However, they do not converge to the true solution of continuum mechanics as resolution is increased, stiffness and damping constant are hard to configure since they are not physical material properties, and volume conservation is hard to enforce [106]. The Finite Element Method (FEM), on the other hand, provides a physical framework closely following continuum mechanics and where material properties can be looked up in the literature. Although deformations are entirely physically based, and hence more realistic and controllable than for mass-spring systems, the computational cost is much higher, especially for non-linear deformations [106]. Mesh-free systems follow continuum mechanics as well, with shape functions used to retrieve a given quantity at a given point in space through the interpolation of known values from neighboring particles. This approach has the advantage of not requiring remeshing during tasks inducing topology changes, such as cutting.

1.2.3 Fluids

Although fluids have received a lot of attention from the computer graphics community, it is not the case in the haptics community. However, fluids are an important component of VE, when used, for instance, in haptic surgical simulators to simulate blood.

The possibilities of haptic interaction with fluids have been explored in relatively recent work, but only in the context of kinesthetic feedback. To the best of our knowledge, no vibrotactile rendering approach has been reported. Different techniques were used to lower the computational demands of the simulation in order to achieve acceptable frame rates for haptic interaction. The Navier-Stokes equations, the laws governing the behavior of fluids, are described in Appendix A. When discretizing these equations, two approaches stand out: the Eulerian approach, where fluid behavior is computed at a number of fixed points in space, with high precision but high computational load, and the Lagrangian approach, where the fluid is discretized into particles that move in space, with overall less accuracy but faster to compute. Surprisingly, only the Eulerian approach (described in Appendix A) has been used for haptic feedback computation.

A precomputed approach is introduced by Dobashi *et al.* [25], where the non-linear interaction forces between fluid and rigid bodies are precomputed. These forces are added to the linear forces computed in real-time, hence obtaining very high frame rates (around 500Hz), since the main computations are performed offline. Some results are shown in Figure 1.21. However, since force models are ad-hoc, it dramatically limits the number of different interaction scenarios that can be simulated. The method is illustrated through a canoe simulation. Another approach is the extension of a 2D technique to 3D space, introduced by Mora and Lee [107]. The fluid surface is modeled as a 15x15 mass-spring network, with a stack of 2D fluid layers below the surface. The layer deformations produced by the proxy are transmitted to the other layers according to their depth and density. Since the algorithm is intended for low-end computers with a small computing power, there is a trade-off between the realism of the simulation and its speed, as well as a limited number of degrees of freedom.

Other techniques are based on existing Eulerian physically based simulations of fluids. Baxter and Lin [27] propose a 6DoF interaction technique with an Eulerian viscous fluid. The two-way rigid-fluid interaction is achieved by discretizing the boundary of the proxy into the simulation grid (Figure 1.22, right). Forces and torques exerted on the proxy are computed by summing the contributions of each edge of the discretized proxy boundary. The authors illustrate the technique through a painting application, as shown in Figure 1.22 (left), with the fluid defined locally around the brush, hence allowing the painter to feel the paint, and achieving 40 to 70 frames per second (fps). However, interaction possibilities are limited due to the high computational cost of Eulerian approaches, requiring the use of simple proxy shapes and small amounts of fluid. Another haptic interaction implementation with an Eulerian fluid has been recently shown by Yang *et al.* [26], where the simulation is accelerated through GPU. The user is allowed to interact with smoke via a spherical proxy at 30 to 75 fps, as shown in Figure 1.23. However, the viscosity of the fluid is not taken into account, and the implementation is limited to 3DoF with simple object shapes, considerably reducing the interaction possibilities.

The models presented so far were specifically designed for the generation of kinesthetic or vibrotactile feedback. However, many VR simulations involve more than the visual and

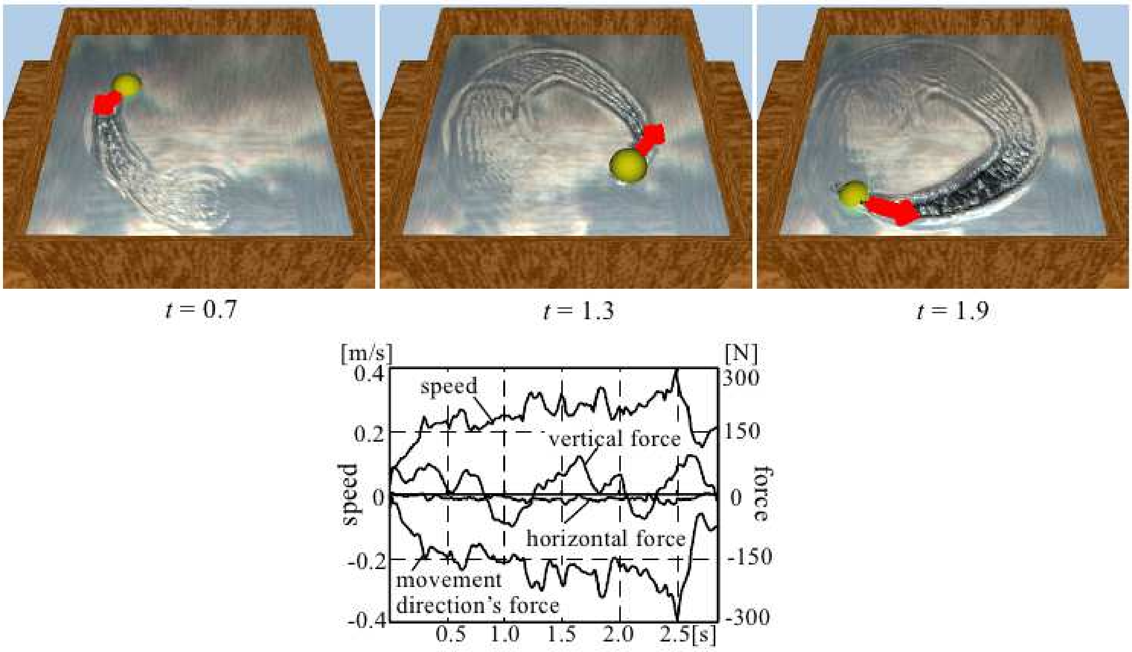


Figure 1.21 – Precomputed haptic interaction with fluids. Top: interaction with a volume of fluid through a sphere. The red arrow indicates the direction of force feedback. Bottom: the corresponding speed and force plots [25].

haptic modality. In the next section, we overview models and architectures that combine haptics with other modalities.

1.3 Combining Haptics with other modalities

The use of haptics in a VR simulation not only improves user immersion by providing an important feedback modality, but can also improve user performance for a given task [14, 15, 42]. Thus, haptics play a fundamental role in bringing VR systems closer to fully immersive environments. However, simulating and rendering the haptic modality alone is pointless, excepts perhaps for very specific scenarios. Indeed, as in real life, without other sensory modalities, and mainly without the visual one, the surrounding VE is very hard to perceive.

In many everyday scenarios, vision outweighs all other senses. Imagine exploring an environment with the lights off, compared to an exploration with earplugs or without physical contact. Thus, visual feedback is fundamental: most VR simulations perform, at least, the visual rendering of the environment. The acoustic modality is also commonly used, as it does not require sophisticated or expensive equipment. VR setups often include an array of speakers producing spatialized sound. Olfactory and gustatory feedbacks are quite challenging to synthesize, but are receiving increasing attention from the VR community [108, 109].

Several studies have quantitatively and qualitatively evaluated the effects of using multiple modalities in VR for a given task [110, 111, 112, 113]. Visual, haptic and auditory feedback are usually considered. In general, the studies showed that the combination of two or more modalities improved user performance for the completion of the specific task.

Since multimodal experiences present many advantages compared to unimodal ones,

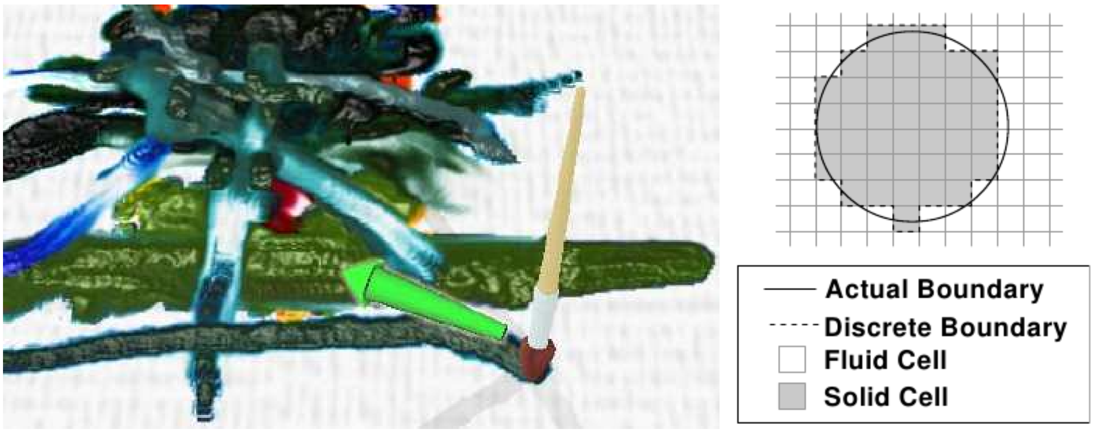


Figure 1.22 – Physically based haptic interaction with viscous fluid. Left: visual rendering of a painting application using the fluid simulation and haptic feedback. The green arrow indicates the force feedback direction. Right: proxy discretized into the Eulerian grid [27].

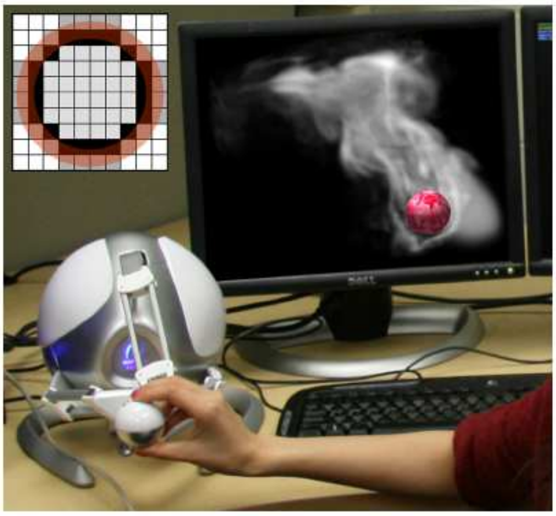


Figure 1.23 – Physically based haptic interaction with smoke. Visual rendering of the application with a proxy discretization into the Eulerian grid (top left) [26].

different approaches have tried to integrate haptics with other modalities. We have chosen to divide the literature in two categories. First, low-level integration approaches, where the same simulation algorithm is used for two or more modalities. This is typically the case of vibrotactile and acoustic feedback, due to vibrotactile and acoustic phenomena common physical generation mechanisms and sources. Second, high-level integration approaches, where the simulation and rendering for each modality is done by a specific algorithm, but within a common simulation or rendering architecture.

1.3.1 Low-level integration

Multimodal integration at low-level allows the use of a single simulation model for the rendering of several modalities.

The CORDIS-ANIMA [19] simulation framework allows, to some extent, the multimodal rendering of a simulation. This framework, originally developed for the music and sound synthesis community, uses a mass-link formalism to model different virtual objects. Material elements are linked together using link elements, which compute the different interaction forces between the material elements using specific algorithms. Classic links are spring and friction elements, but other links have been designed, exhibiting, for instance, more complex non-linear behavior. The differential equations governing the simulation are solved using finite differences. The outputs of the simulated system (positions, forces, vibrations, etc) can be sent to physical transducers (audio, haptic and visual devices) to be perceived by the user. Through this framework, several virtual music instruments and sound effects have been designed [114], and many different materials have been simulated to some extent [115]. Vibrations can be extracted from the simulation to be presented as audio feedback, as well as vibrotactile feedback. Kinesthetic feedback has also been addressed through direct rendering and specific devices [116].

More recent work has focused on the use of physically based approaches relying on the same principle: using a common model to generate acoustic and vibrotactile feedback, due to the common physical source of both phenomena in an interaction context. Visell *et al.* [31] use a fracture model to simulate snow, gravel and sand, which is rendered through tiles equipped with vibrotactile transducers. Using the same devices and model, the cracking ice of a frozen pond [32] is also displayed, as shown in Figure 1.24. This work is described in the section addressing granular materials 1.2.2.2.a. The visual rendering algorithms are ad-hoc techniques for each simulated material. Papetti *et al.* [17] use various acoustic models from the Sound Design Toolkit [117] to generate the vibrations of crumpling materials. The signals are displayed through custom shoes equipped with loudspeakers and haptic actuators. Nordahl *et al.* [30] developed a similar approach with rigid, friction and aggregate models. They conducted a material recognition experiment, showing that some material properties could be conveyed through haptic rendering alone.

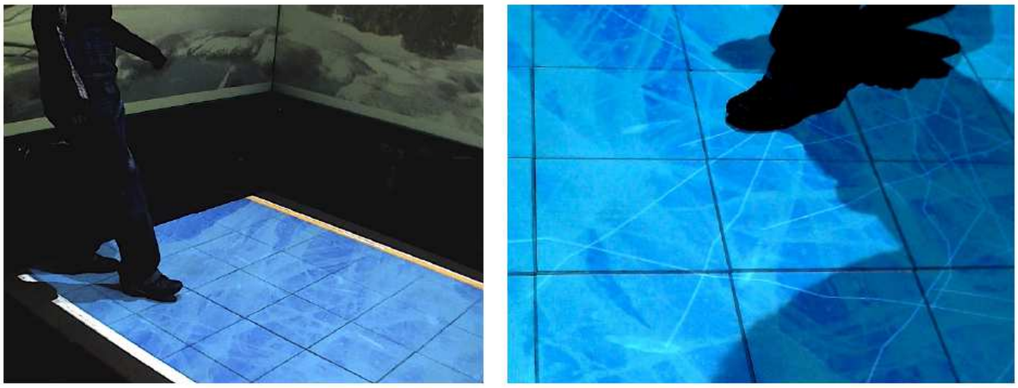


Figure 1.24 – Example of a multimodal low-level integration. The frozen pond scenario generates vibrotactile and acoustic rendering through a fracture model [32].

1.3.2 High-level integration

Multimodal integration at high-level use different models, one for each modality, and combines them in a single architecture. Numerous multimodal rendering architectures combining visual, auditory and haptic feedback were developed. These architectures are found in common application areas of VR, such as virtual assembly and prototyping, medical simulators and education.

In the area of virtual assembly and prototyping, early architectures by Gupta *et al.* [118] use a desktop environment with stereoscopic rendering, two Phantom 3DoF devices, and prerecorded contact sounds for a peg-in-a-hole task. Haptic rendering uses penalty forces and a simplified friction force model. An evaluation comparing the virtual task with the real one showed that task completion time is higher in the virtual case. Baier *et al.* [119] follow a similar approach for multimodal telepresence (drilling application) and virtual presence (simple ping pong example). Haptic feedback is computed on simple models (rigid or deformable through mass-spring systems), using penalty forces and output to a 3DoF device. A main difference is the distributed nature of computations, with different computers for each modality. Instead of using precomputed sound, they synthesize sounds in real-time based on a precomputed Fourier analysis of real contact sounds.

More advanced multimodal architectures for virtual prototyping allow the interaction with more complex virtual object, with a high polygon count. In MIVAS [120], the more efficient voxel-based Voxmap-PointShell [80] approach is used for haptic feedback in a virtual assembly task. A CAVE [121] setup is used, together with a finger tracking and force feedback device. Acoustic feedback is rendered using prerecorded contact and environment sounds. Diaz *et al.* [122] also use a voxel-based approach for collision detection, but rely on direct rendering through penalty forces for haptic feedback on a 3DoF haptic device. A slow thread (30Hz), with simulation and graphic rendering loops, runs on one machine, while a faster (1kHz) audio and haptics thread runs on another machine (fast thread). Data is interpolated and extrapolated between the two threads due to the use of different frequencies. Stereoscopic rendering is used for the visual modality, and sound is generated using real-time modal audio synthesis [123] using predefined modal frequencies for materials such as metal, plastic and wood. The application is shown in Figure 1.25. A subjective study showed that multimodal feedback improved user performance.

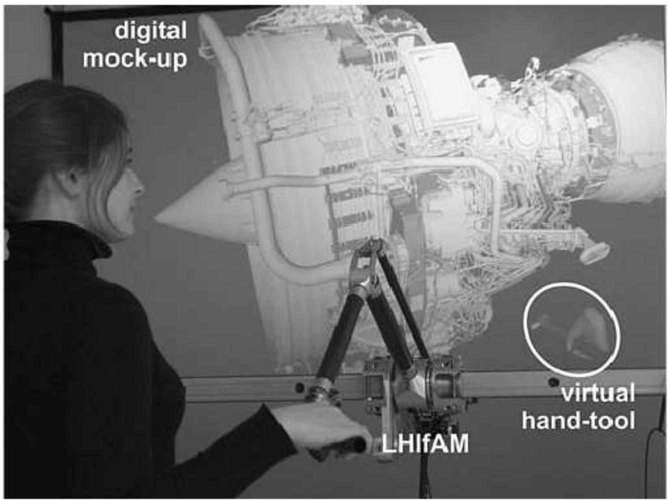


Figure 1.25 – High-level multimodal integration for virtual prototyping: a distributed approach with visual, acoustic and kinesthetic feedback [122].

The most elaborate architecture for virtual prototyping is presented by Sreng *et al.* [2] (Figure 1.26), designed for the computation of multimodal feedback of contact events. Visual, acoustic and haptic (both kinesthetic and vibrotactile) feedback are computed from different impact, detachment and friction states and events. Particle-based effects are used for the visual rendering of events. For acoustic rendering, a precomputed modal analysis is used to create a modal resonator filter. Then, a Gaussian-like excitation signal for impact and a noise-like excitation signal for friction are fed to the modal filter [124]. The resulting signal goes through a sound spatialization post-processing step before being rendered. Kinesthetic feedback uses virtual coupling and a constraint-based approach for force computation, with the superposition of open-loop rendering of contact events as described in section 1.2.1.2.b.

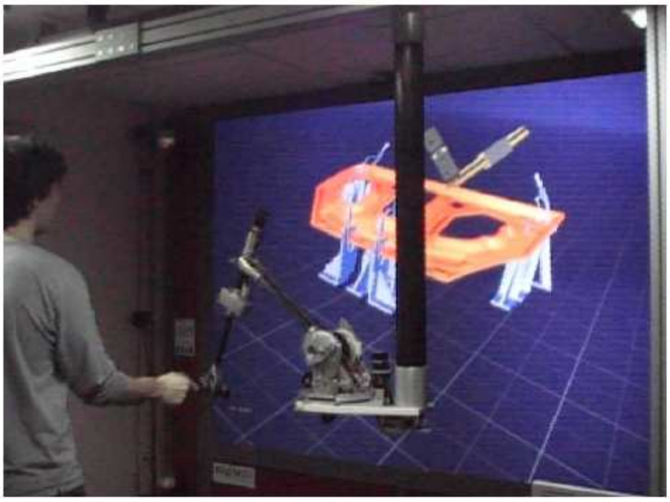


Figure 1.26 – Elaborate high-level multimodal integration: physically based acoustic, kinesthetic and vibrotactile feedback are rendered with the usual visual display in a virtual prototyping and assembly context [2].

Several multimodal examples can also be found in the medical field for training applications. The Munich Knee Joint Simulation project [7] (Figure 1.27) enables the virtual diagnosis of knee joint lesions with multimodal feedback. A desktop-based visual feedback is combined with the acoustic display of prerecorded noise from inner structures of the knee joint and prerecorded sounds of the patient in pain. The thigh of a fake human is attached to a grounded haptic device, which can move the knee joint in 6DoF with predefined loads and angles. Another example in the area of medical systems is bone drilling [8]. A 3D skull model obtained from computed tomography data and a medical drill model are displayed in a desktop environment. An ad-hoc haptic mechanism based on the correctness of user manipulation provides force feedback to the user. Three different recorded sounds can be presented to the user, corresponding to the sound made by the real drill in three cases: not enough force applied, correct drilling, and too much force applied. A subjective study showed that multimodal feedback improved user performance for the drilling task. Wan *et al.* [6] present a dental training system, with 3DoF haptic rendering through penalty forces and direct rendering. For acoustic feedback, spectral analysis is used to extract the main frequencies from proxy-tooth real impact sounds. Then, signal synthesis is used on impact using linearly attenuated sinusoidal signal using the extracted frequencies and white noise to simulate the environment sound. Again, multimodal feedback seems to improve user performance for this particular task.

In a different context, the VIREPSE multimodal VE [125] uses a rather unusual approach by adding olfactory feedback to the previously mentioned modalities. A stereoscopic environment is combined to a SPIDAR-like device for force feedback and prerecorded sounds for the exploration of the abstract concept of the Bohr atomic model. Atomic energy levels are represented as spheres, providing multimodal feedback on contact. Besides the surprise and fun factor of the olfactory feedback, users questioned its usefulness in this sort of application.

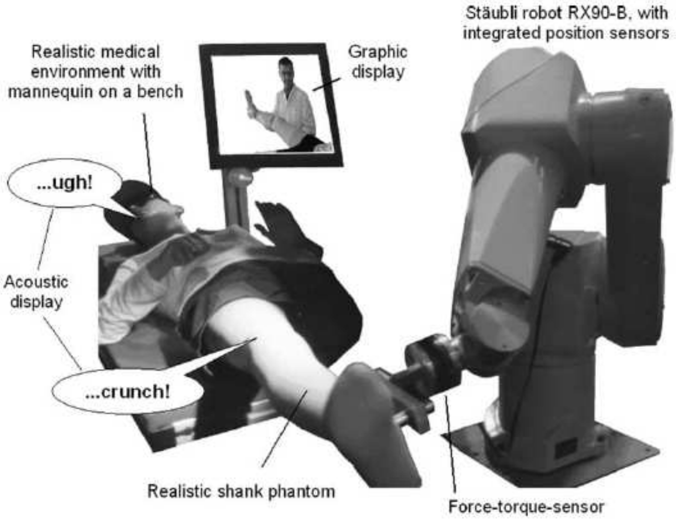


Figure 1.27 – The Munich Knee Joint Simulation project: virtual diagnosis of knee joint lesions with visual, acoustic and haptic feedback [7].

1.4 Conclusion

In the perspective of studying the haptic and multimodal manipulation of physically based complex VE, this chapter presented a synthetic survey of the fundamentals of haptic interaction, and the different models for haptic and multimodal interaction with physically based rigid, deformable and fluid media. We have first shown that the haptic modality involves a human component (the human haptic system), and a machine component including haptic devices as the physical interface between users and VE, and a set of rendering algorithms on the simulation side.

In a VR context, we have distinguished two important haptic rendering categories, different in terms of human perception, type of device and rendering architecture: kinesthetic rendering, involving body positions and forces, and vibrotactile rendering, involving higher frequency vibrating signals at the surface of the skin. We surveyed the existing haptic rendering techniques allowing the interaction with different media along these two categories. We encompassed the three types of media through which most phenomena can be simulated: rigid bodies, deformable bodies and fluids. Finally, we showed how previous research has developed or integrated different models allowing the combination of haptic feedback with other modalities such as vision and hearing, leading to multimodal applications.

Through this overview, and considering the objectives of this Ph.D. thesis, several issues stand out.

First, the haptic rendering of fluids has been scarcely studied. Existing approaches do not take full advantage of the 6DoF interaction possibilities offered by today's devices, and are limited to simple shapes and small amounts of fluid. Rich haptic experiences, such as holding fluid through concave objects, are not possible. One of the main reasons is the use of an Eulerian fluid simulation, leading to expensive computations and complex or limited solid-fluid coupling mechanisms. Surprisingly, these techniques do not focus on Lagrangian particle-based models, which have proven to be fast to simulate and yield good results in the graphics community. In addition, the Lagrangian approach is more intuitive in the way it treats matter, and is therefore perhaps better suited for haptics than the Eulerian approach. These issues are addressed in Chapter 2, when we present a 6DoF haptic rendering technique for the interaction with fluids.

Second, we can see in this chapter that the haptic interaction with a VE is achieved in many different ways, with different haptic rendering approaches and different physically based models. However, these differences are mainly due to the existence of different types of media (rigid, deformable, fluid): no haptic rendering technique works on all three states indistinguishably. Existing approaches are not designed to encompass the three states at the same time, due to different underlying physical models, different contact representations and different data-structures. Thus, we can see, at most, two different interacting states, such as rigid-deformable or rigid-fluid interaction. This issue is addressed in Chapter 3, when we propose a unified approach for the 6DoF haptic interaction with rigid, deformable and fluid media.

Third, we can see that the body of literature on physically based vibrotactile rendering is rather small, particularly in the non-rigid case. This is due to the inadequation of the vibrotactile modality to low-frequency deformations. However, the potential of vibrotactile rendering in VR remains high, thanks to the availability of cheap and generic devices. We can also observe that most techniques addressing vibrotactile rendering of non-rigid media are inspired from acoustic models due to the common physical source of both phenomena. This approach allows the low-level integration of different modalities, bringing compelling multimodal results without requiring different models. We address the vibrotactile rendering of non-rigid media in Chapter 4, leveraging this sound-based approach in a novel context where no previous work has been reported: the vibrotactile rendering of fluids.

Six Degrees-of-Freedom Haptic Interaction with Fluids

2

Contents

2.1 Smoothed-Particle Hydrodynamics Fluid Simulation 46
  2.1.1 SPH Discretization 47
  2.1.2 SPH Fluid Model 47
2.2 3DoF Haptic Rendering of Fluids 48
  2.2.1 Smoothing Volume and SPH Haptic Forces 48
  2.2.2 Haptic Rendering Algorithm 49
  2.2.3 GPU Implementation 49
2.3 6DoF Haptic Rendering of Fluids through Rigid Body Interaction 51
  2.3.1 Unified Particle Model 51
  2.3.2 Rigid Body Dynamics 53
  2.3.3 6DoF Haptic Coupling Scheme 54
  2.3.4 Virtual Coupling 55
2.4 Visual Fluid Rendering 55
  2.4.1 Computing Per-Pixel Fluid Data 55
  2.4.2 Fluid Compositing 56
2.5 Evaluation 56
  2.5.1 Hardware Setup 56
  2.5.2 Computation Time 57
  2.5.3 Graphic Rendering 58
  2.5.4 Example Scenarios 58
  2.5.5 A complete use-case 61
2.6 Discussion 61
2.7 Conclusion 63

The addition of haptic feedback to a VR application is a major improvement when simulating real life scenarios, in terms of immersion and performance [15, 42]. Most current haptic simulations involve only rigid or deformable bodies. Fluids, on the other hand, have been scarcely studied, while being broadly present in our daily life, as well as in industrial and medical contexts. However, achieving realistic, stable and real-time force feedback from fluids is particularly challenging. To simulate interactions between fluids and rigid bodies within haptic rendering, previous studies have proposed precomputed ad-hoc algorithms [25], approaches featuring only 3DoF and non-viscous fluids [26], or implementations restricted to simple object shapes and small amounts of fluid [27]. Thus, as for today, there is a lack of models and rendering techniques handling complex 6DoF haptic interactions with viscous fluids in real-time.

In this chapter, we propose the first approach to enable real-time 6DoF haptic interaction with viscous fluids through arbitrary shaped rigid bodies and 6DoF haptic devices. This represents a significant leap forward in interaction possibilities compared to previous work on haptic interaction with fluids. Until now, real-time haptic interaction was restricted to simply "swiping" the surface of a fluid volume with simple objects, dramatically reducing the interaction possibilities when compared to real-life scenarios. Our fluid haptic rendering technique is based on Smoothed-Particle Hydrodynamics [28] (SPH) for haptic and fluid simulation [29], and is the first attempt to bring the SPH model to the haptic fluid realm.

This chapter is organized as follows. Section 2.1 recalls the main equations involved in the SPH simulation of fluids. Building upon SPH, we describe our novel haptic rendering technique, initially coupled to a single particle and with 3DoF haptic feedback, in section 2.2. The algorithm is extended in section 2.3 for 6DoF haptic rendering through the use of a unified particle model for rigid body interaction with fluids. Section 2.4 describes our screen space fluid graphic rendering algorithm. We evaluate and illustrate the main features of our approach through different scenarios in section 2.5, highlighting the 6DoF haptic feedback and the use of containers. We discuss our results and the limitations of our technique in section 2.6, before concluding.

2.1 Smoothed-Particle Hydrodynamics Fluid Simulation

Two main fluid simulation approaches based on Navier-Stokes equations can be distinguished [126]: the Eulerian approach, where space is subdivided in a grid and the different physical quantities are computed for each grid cell, and the Lagrangian approach, where the fluid is treated as a system of independent particles, each with a position and a velocity. We focus on the Lagrangian approach, since the simulated fluid is not bounded by a grid and is much faster to compute, which is especially interesting for haptic feedback. On the other hand, compared to the Eulerian approach, Lagrangian fluids have a lower order spatial accuracy, enforcing incompressibility is quite challenging and some interesting behaviors can disappear in low-density regions.

SPH are the preferred choice when simulating Lagrangian fluids in the graphics community. Thus, in this section we recall the relevant equations involved in the SPH simulation of fluids, for the computation of the physical quantities (density) and the related forces (pressure and viscosity forces). For further details, we refer the reader to [29] and [126].

SPH were developed by Lucy [127] and Gingold and Monaghan [128] for the simulation of astrophysical phenomena. They were introduced in the graphics community by Stam and Fiume [129] for gaseous phenomena, and Desbrun and Cani [130] for highly deformable bodies. Müller *et al.* [29] proposed the first approach for fluid simulation using SPH.

An SPH fluid simulation is based on particles carrying different physical properties, such as mass and viscosity, discretizing the fluid volume. These particles have a smoothing radius, a spatial distance defining a neighborhood around them. Physical quantities, such as density and interaction forces, can be computed for each particle through the weighted sum of the relevant properties or quantities of the particles inside its neighborhood. The weight of the contributions of a neighbor particle depends on its distance to the treated particle, and is defined in a function called smoothing kernel.

2.1.1 SPH Discretization

The smoothed quantity Q_i of a particle i at any position $\mathbf{x}_i$ in space is computed through the general formula:

$$Q_i = \sum_j Q_j V_j W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{2.1}$$

where Q_j is the discrete quantity Q sampled for neighboring particle j at position $\mathbf{x}_j$, V_j is the volume of j, and W is the smoothing kernel of support h, where particles farther than the distance h are not taken into account.

Hence, the density ρ of particle i can be computed through:

$$\rho_i = \sum_j m_j W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{2.2}$$

where m_j is the mass of neighboring particle j. We recall $V_j = m_j/\rho_j$, with ρ_j being the density of particle j.

2.1.2 SPH Fluid Model

The motion of fluids is driven by the Navier-Stokes equations, with the following formulation for Lagrangian simulation:

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla P + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v} \tag{2.3}$$

where P is the pressure, $\mathbf{v}$ the velocity, μ the viscosity coefficient and $\mathbf{g}$ the gravity field. ∇ and ∇^2 are respectively the gradient and Laplacian of the physical quantities. This formulation leads to 3 distinct forces:

$$\rho \mathbf{a} = \mathbf{f}^{pressure} + \mathbf{f}^{gravity} + \mathbf{f}^{viscosity} \tag{2.4}$$

with $\mathbf{a}$ being the particle acceleration.

Pressure forces are computed from pressure quantities, and viscosity forces are computed from velocities:

$$\mathbf{f}_i^{pressure} = -V_i \sum_j V_j \frac{P_i + P_j}{2} \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{2.5}$$

$$\mathbf{f}_i^{viscosity} = \mu V_i \sum_j V_j (\mathbf{v}_j - \mathbf{v}_i) \nabla^2 W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{2.6}$$

The kernel functions and the symmetrization of the force computations follow those proposed in [29]. The pressure computation follows the modified ideal gas state equation proposed in [130]:

$$P = k(\rho - \rho^0) \tag{2.7}$$

where k is a gas constant that depends on the temperature, and ρ^0 the rest density.

Therefore, the fluid simulation is done in two consecutive steps. First, the density is computed for each particle, and the pressure is derived from it. Second, after computing pressure and viscosity forces, acceleration, velocity and position are deduced through a Leap-Frog explicit integration scheme [131].

Parameters such as the viscosity can be changed to obtain different fluid behaviors, from smoke (no viscosity) to very high viscosity fluids such as honey.

2.2 3DoF Haptic Rendering of Fluids

In this section, we describe our novel haptic rendering algorithm based on Smoothed-Particle Hydrodynamics for 3DoF haptic feedback. Our method allows to compute force-feedback when interacting with fluid through a single fluid particle. Our technique can be classified in the group of distance-field-based haptic rendering techniques, and can therefore be used in simulations where exact surface representation is not required [50]. This is the case for fluids, and particularly SPH fluids, where the exact boundary of the fluid is not explicitly defined: boundary particles are not the boundaries of the fluid, but rather the last points in space that provide information about the fluid boundaries.

2.2.1 Smoothing Volume and SPH Haptic Forces

In the SPH model, forces are computed when a particle enters the smoothing volume of another particle, which is defined as a sphere with the smoothing radius as radius and the particle position as center. The sum of the smoothing volumes of the particles of an entity defines the Smoothing Volume of the entity, behaving like a force field volume around the entity. The Smoothing Volume is illustrated in Figure 2.1. Any external particle (not belonging to the entity) inside the entity Smoothing Volume will trigger the computation of forces exerted on both the entity and the external particle.

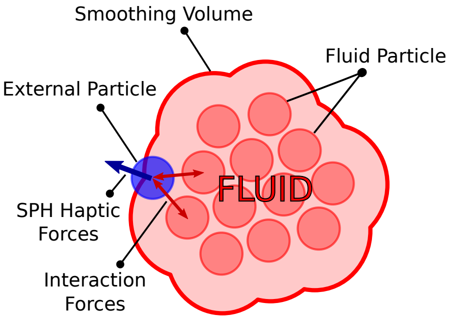


Figure 2.1 – Smoothing Volume and SPH haptic forces. The Smoothing Volume of a volume of fluid is defined by the sum of the smoothing volumes of each particle belonging to the fluid. When an external particle enters the Smoothing Volume, SPH haptic forces are seamlessly computed between the external particle and the fluid particles whose smoothing volume contains the external particle.

The forces generated when an external particle is inside the Smoothing Volume of an entity are SPH haptic forces, noted $\mathbf{f}^{haptic}$. They repel the external particle from the entity, and the entity from the external particle in a 2-way coupling scenario. Following the computation of quantities in the SPH model, $\mathbf{f}^{haptic}$ has the general form of Equation 1.

The profile of the haptic rendering is closely related to the choice of $\mathbf{f}^{haptic}$. When interacting with fluid through a particle, or in other words when the haptic device is coupled to a particle, an intuitive choice is to treat the coupled particle as a fluid particle. The coupled particle is submitted to the same physical forces as any other fluid particle. Since these forces are then rendered through the haptic device, they provide a direct way to interact with the fluid volume. Hence, our SPH haptic forces, when coupled to a particle, are the sum of pressure and viscosity forces:

$$\mathbf{f}_p^{haptic}(\mathbf{x_p}) = \mathbf{f}_p^{pressure}(\mathbf{x_p}) + \mathbf{f}_p^{viscosity}(\mathbf{x_p}) \tag{2.8}$$

2.2.2 Haptic Rendering Algorithm

Algorithm 1 shows the different steps of the computation of haptic feedback when interacting with fluid through a single particle coupled to a haptic device. It can be broken down to 4 distinct steps:

- **density computation (lines 1-3)**. A density quantity is computed for each particle, according to the neighboring particles distance and mass.
- **SPH haptic force computation (lines 4-12)**. According to the type of particle (a flag specifying fluid or external), the algorithm computes either fluid interaction forces from section 2.1.2, or SPH haptic forces from section 2.2.1.
- **fluid particle integration (lines 13-16)**. After adding external forces (such as gravity), a new position and velocity is computed for each fluid particle by integrating forces over the simulation time step.
- **coupled particle integration and haptic coupling (lines 17-22)**. After adding external forces (such as gravity), particles coupled to a haptic device follow a different integration process, taking into account forces from the haptic interface, and sending back their new position and velocity.

2.2.3 GPU Implementation

Simulating physically based fluids is a computationally expensive task, a critical factor for real-time applications. Full GPU implementations include the work of Harada *et al.* [132], where they use a texture representation of a grid space subdivision for neighbor computation purposes, and Zhang *et al.* [133], where values are accumulated by computing the contribution of each particle *to* its neighbors, instead of computing the contribution *from* its neighbors. Although efficient, these approaches are either hard to implement, or require a huge memory consumption.

In our approach, the entire physical model is implemented on GPU using the CUDA framework [134]. It is based on Green's implementation of a CUDA particle simulator [135], modified to compute the SPH Navier-Stokes equations. Our approach has the advantage of being simpler than previous ones, while providing the same efficiency.

2.2.3.1 Optimizations

The increase in performance compared to a CPU implementation comes from the parallelization of the density, force and integration computations. Knowing that an SPH particle only interacts with other particles inside its smoothing radius, the 3D interactive space is subdivided in a regular grid, restricting the search for neighbors to the 26 grid cells around the particle cell and the particle cell itself. Major optimizations regarding memory access and data structures [135] are:

- the sorting of the particles by cell in GPU memory in order to ensure a coalesced read, hence optimizing memory access;

Algorithm 1 Haptic rendering algorithm for a single particle coupling: the different steps of the computation of haptic feedback from the interaction between fluid and external particles are illustrated. Each external particle is coupled to a haptic device.

```
for all particles in fluid do
  compute density (Eq. 2.2)
end for
for all particles in fluid do
  for all neighboring particles do
    if neighbor is coupled to a haptic device then
      compute SPH haptic forces (Eq. 2.8)
    else
      compute fluid forces (Eqs. 2.5 and 2.6)
    end if
  end for
end for
for all particles in fluid do
  add external forces (gravity)
  integrate (new position and velocity)
end for
for all particles coupled to a haptic device do
  add external forces (gravity)
  add force from haptic interface
  integrate (new position and velocity)
  send new position and velocity to haptic interface
end for
```

- the binding of the sorted data arrays to textures, and the use of texture lookups to benefit from texture data caching;

- the use of an OpenGL Vertex Array Object for the particle position array, allowing the graphic rendering algorithm to access position data without copying it back to CPU memory.

2.2.3.2 Update rate

In theory, the graphic rendering frame rate and the update rate of the haptic device could and should be independent. Graphics should be rendered at 24 fps for a comfortable visualization, while the haptic update rate, depending on the computation time of the fluid simulation loop, requires a higher frequency. However, using an all-GPU design to achieve plausible haptic interaction rates requires the use of the GPU for simulation computations *and* for graphic rendering. In current implementations of graphic drivers, graphic rendering is a GPU blocking task, hence the simulation and the graphic rendering on the GPU are not parallelizable. Hence, using a different frequency for graphic update would cause periodic drops in the haptic update rate, introducing artifacts in the haptic rendering. Since every simulation loop has to be rendered, we needed to optimize our graphic rendering algorithm to achieve reasonable haptic update rates, as detailed in section 2.4. A possible way to dissociate the graphic from the simulation loop would be the use of two GPUs, with one doing simulation computations and the other focusing on graphic rendering. The position data required for the graphic rendering would be copied from the simulation GPU to the graphic rendering GPU. This approach is presented in Chapter 3. In addition, the advent of multi-core GPUs would make dissociation even simpler by removing the need of data transfer between GPUs [136] [137].

2.3 6DoF Haptic Rendering of Fluids through Rigid Body Interaction

In the previous section, we showed how to generate force feedback when interacting with fluid through a single particle. In this section, we extend our haptic rendering technique to allow the interaction between a rigid body and the fluid, hence producing a 6DoF haptic feedback when coupled to a 6DoF haptic device. We describe our rigid body model, the computation of rigid body dynamics using SPH particles, and the underlying haptic coupling scheme. The approach is illustrated in Figure 2.2.

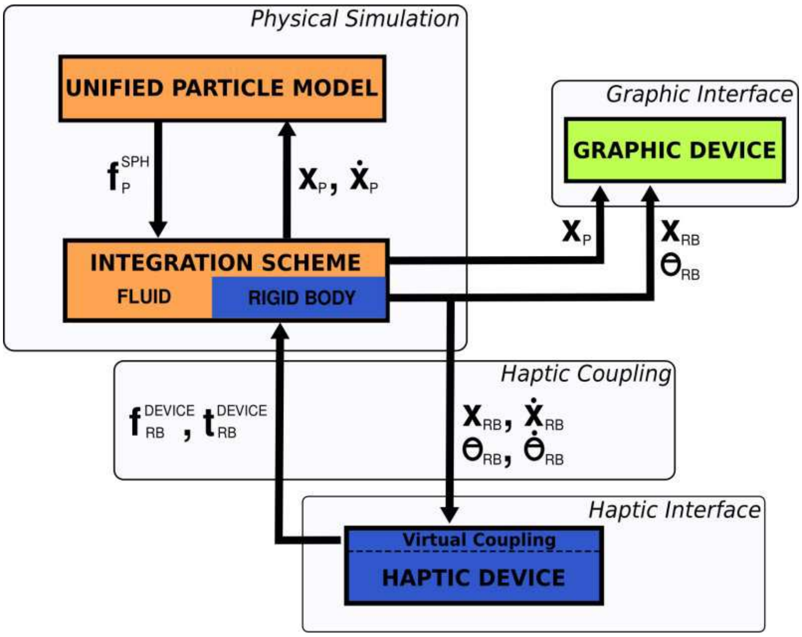


Figure 2.2 – Overview of the 6DoF haptic rendering of fluids. The unified particle model computes the different interaction forces between SPH particles P, $\mathbf{f}_P^{SPH}$, including haptic feedback forces. Then, rigid body dynamics are applied to the coupled rigid body RB, taking into account the force $\mathbf{f}_{RB}^{DEVICE}$ and torque $\mathbf{t}_{RB}^{DEVICE}$ generated by the 6DoF haptic device through Virtual Coupling. The new linear and angular velocity ($\dot{X}_{RB}$, $\dot{\theta}_{RB}$) and position (X_{RB}, θ_{RB}) are sent to the haptic device.

2.3.1 Unified Particle Model

In order to achieve 6DoF haptic interaction with fluids, we required a rigid body model that could fit in the SPH simulation with a minimum impact regarding computation time, and that would allow a flexible coupling mechanism. We propose a real-time approach based on [138] and improved for haptic interaction by bringing it to high speeds: a unified particle model allowing the seamless real-time interaction between fluid and arbitrary-shaped rigid bodies.

Rigid bodies can be simply and efficiently modeled with the same SPH particles used in the fluid simulation. This allows to use the SPH model for the computation of forces between fluid and rigid body particles, removing the need of additional collision detection algorithms. In addition, since interactions are computed between particles, the overall shape of the rigid body is not important. Hence, the unified particle model allows the seamless use of arbitrary-shaped rigid bodies, including concave objects.

Rigid body polygonal meshes are sampled into a set of particles covering the surface of the mesh. Figure 2.3 shows a bowl mesh and its corresponding particle sampling. Several mesh sampling techniques exist in the literature. We highlight the work presented in [139], for an offline simulation, where the particle sampling is achieved in a preprocessing step using a distance field, and the random placement of particles inside the distance field. Particles are then subjected to an attraction constraint to the zero-isosurface and a repulsion constraint against other particles, achieving an efficient sampling of the surface when reaching a convergence criteria. The survey and the improvement of other techniques are, however, beyond the scope of this chapter.

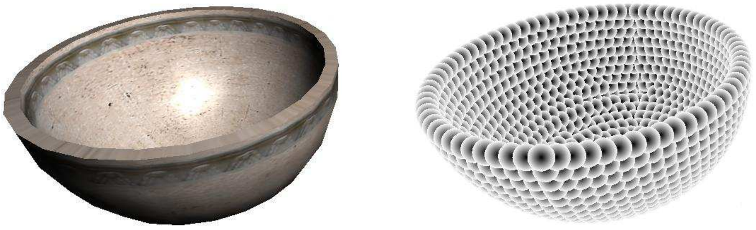


Figure 2.3 – Rigid body particle sampling. The mesh bowl (left) is converted into a set of particles (right).

The SPH fluid simulation and the haptic rendering algorithm are adapted to allow the simulation of rigid bodies and their interaction with the fluid. These changes are:

- rigid body particles interact with fluid particles through the SPH haptic forces $\mathbf{f}^{haptic}$;
- rigid body particles are given constant densities, since for rigid bodies these quantities are constant throughout the simulation;
- when computing the density of fluid particles, rigid body particles are omitted so that fluid densities are only computed with fluid particles;
- rigid bodies follow rigid body dynamics, as described in section 2.3.2.

The choice of an efficient $\mathbf{f}^{haptic}$ for a rigid body coupling case is not necessarily the same as in single particle coupling. In the previous section, the coupled particle could be treated as a fluid particle, and hence $\mathbf{f}^{haptic}$ was set to follow fluid forces. When dealing with rigid bodies, previous work has used different forces and mechanisms to simulate a solid-fluid interaction. Among these methods are simple penalty forces [140], Lennard-Jones forces [141] and unified fluid-rigid body forces [142] [138].

In a unified, parallel and time-critical framework, unified fluid-rigid body forces fit well for computation time reasons. Using the same interaction forces as in a fluid-fluid case improves the gain of parallel computation, while providing a reasonable amount of control over the forces through density and viscosity values. These values are set by the user per rigid body, hence allowing different behaviors for each rigid body. However, since Navier-Stokes equations are not physically meant for rigid bodies, density and viscosity values cannot be looked up in the literature, but need to be chosen empirically. This is particularly true for viscosity forces, where higher solid-fluid viscosity values are required to achieve the desired viscous and sticky effects.

We used the pressure forces as for fluid particle interaction, and symmetrized the viscosity forces as in [143] to account for the possibility of having different viscosity values:

$$\mathbf{f}_p^{visc}(\mathbf{x_p}) = V_p \sum_i V_i \frac{\mu_i + \mu_j}{2}(\mathbf{v}_i - \mathbf{v}_p)\nabla^2 W(\mathbf{x_p} - \mathbf{x_i}, h) \tag{2.9}$$

2.3.2 Rigid Body Dynamics

With rigid bodies, position and velocity values are no longer computed independently for each particle. SPH haptic forces due to the interaction with the fluid are summed and applied at the center of mass *cm* of the rigid body, as shown in Figure 2.4. Torques are also applied to the center of mass due to the same SPH haptic forces exerted on the particles of the rigid body.

$$\mathbf{F}_{cm}^{haptic} = \sum_{i_{body}} \mathbf{f}_{i_{body}}^{haptic} \qquad \tau_{cm}^{haptic} = \sum_{i_{body}} \tau_{i_{body}}^{haptic} \tag{2.10}$$

with:

$$\tau_{i_{body}}^{haptic} = (\mathbf{x}_{i_{body}} - \mathbf{x}_{cm}) \times \mathbf{f}_{i_{body}}^{haptic} \tag{2.11}$$

Using Newton's second law we can compute the linear and angular accelerations of the rigid body at its center of mass. The new linear and angular velocities as well as the new position and orientation of the body are then computed using a Leap-Frog integration scheme [131]. The angular velocity is reported to each particle of the body according to their position with respect to the center of mass. We do not provide a detailed explanation of rigid body motion in this chapter. For this, we refer the reader to appendix A.

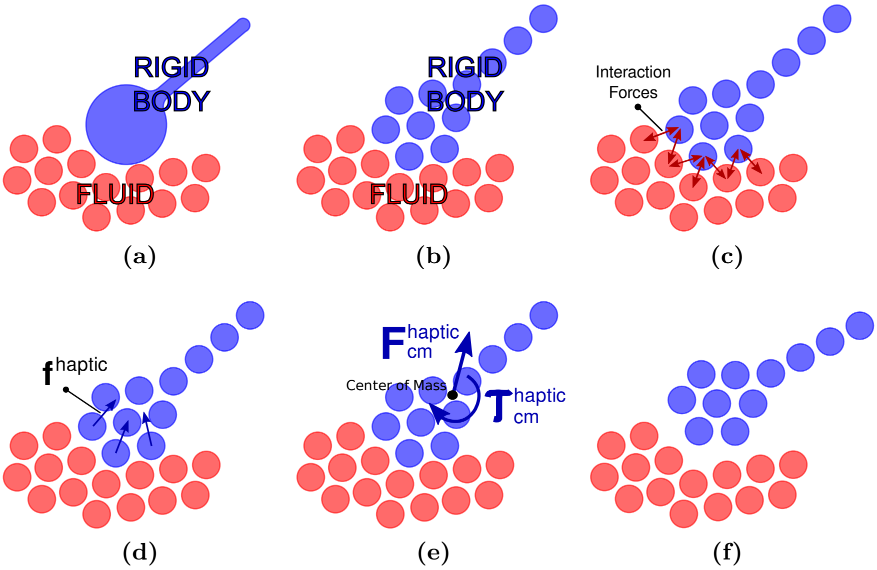


Figure 2.4 – Illustration of the computation of forces acting on a rigid body. A spoon and volume of fluid (a) are modeled with SPH particles (b). Fluid particles act on the spoon particles (c), generating an SPH haptic force $\mathbf{f}^{haptic}$ per spoon particle (d). These forces are summed resulting in a total force and a total torque applied at the center of mass of the spoon (e). Spoon particles are updated according to the new position and velocity of the spoon(f).

2.3.3 6DoF Haptic Coupling Scheme

When coupling a rigid body to a 6DoF haptic device, we have to take into account the new rigid body dynamics. Algorithm 2 shows the new rigid body dynamics step, replacing the external particle integration step from Algorithm 1 (lines 17-22), and performing the 6DoF haptic coupling.

Algorithm 2 6DoF haptic rendering algorithm: rigid body dynamics step and haptic coupling with a 6DoF haptic device.

```
 1: for all rigid bodies do
 2:   for all particles in the rigid body do
 3:     add force exerted on the particle to total force
 4:     add torque from the particle to total torque
 5:   end for
 6:   add external forces and torques (gravity)
 7:   add force and torque from haptic device
 8:   integrate (new rigid body position and velocity)
 9:   send new position and velocity to haptic device
10:   for all particles in the rigid body do
11:     update particle position and velocity
12:   end for
13: end for
```

Figure 2.2 summarizes the different steps of the 6DoF haptic rendering technique, from the unified SPH simulation to the 6DoF haptic loop. Following our unified particle model, SPH haptic forces are computed along other forces in the SPH simulation. In the rigid body dynamics step, all forces $\mathbf{f}^{haptic}$ exerted on the coupled rigid body are summed to obtain a total force and a total torque (lines 1-5 of Algorithm 2). The force and torque feedback coming from the haptic device are added to the coupled rigid body (line 7 of Algorithm 2). Then, the new position and velocity are computed by integrating forces over the simulation time step. They are sent to the haptic device, closing the haptic loop in admittance mode [50].

Rigid bodies are also simulated entirely on the GPU. Rigid body particles are stored in the same arrays as fluid particles, since forces are computed for every particle in the simulation. However, they are grouped at the beginning of the arrays in order to ensure a coalesced memory access during rigid body dynamics computations. The sum of SPH haptic forces into a total force following Equation 12 is computed on the GPU by a 2-step tree-based parallel reduction [144], with sequential addressing. In a first CUDA kernel, the force array is partially summed within blocks of maximum thread size, with results stored in global memory. The second CUDA kernel sums the intermediate results into a final force value. Lines 6 to 8 of Algorithm 2 are executed on a single CUDA thread within the previous kernel, since there was no substantial gain in copying the data and executing the instructions on the CPU.

This coupling scheme allows the seamless haptic coupling with any rigid body, provided dimensions and masses are compatible with the device span and its maximum efforts. It also allows the computation of N different haptic couplings on the same VE, as well as between N devices and the same rigid body (as holding a bucket with two hands).

2.3.4 Virtual Coupling

A Virtual Coupling mechanism [58] is introduced between the haptic device and the coupled rigid body, creating a viscoelastic link between them. This mechanism allows the separation of the impedance of the haptic device from the impedance of the VE. It reconciles a high update rate haptic device with a lower rate simulation, leading to an increase of stability [50].

2.4 Visual Fluid Rendering

As explained in section 2.2.3.2, the haptic loop needs high frequency updates, which required us to design a visual rendering method with performance over quality in mind.

Many techniques have been proposed to render fluids simulated through SPH. To this aim, previous work [145] [146] introduced mesh-based techniques, where the surface of the fluid is extracted at a mesh is reconstructed each time step. These techniques can lead to visually atractive results, but mesh generation is quite expensive, particularly in our haptic context. Recent methods proposed to render fluids in screen space without generating any meshes. In [147], per-pixel normals of the fluid surface in view space are computed using directly the fluid depth buffer, representing the front fluid surface. Thus, per-pixel lighting as well as environment reflection could be computed by rendering a single full screen quad. A similar method has been proposed in [148], where the fluid depth buffer is smoothed out by minimizing its curvature and the overall fluid appearance is improved using high frequency noise. In addition, fluid thickness is taken into account in order to simulate light extinction through the medium.

Our visual rendering is close to the one proposed in [148], with reduced computation time. We propose a three step approach: (1) compute per pixel fluid data (front and back fluid volume depth from view), (2) smooth the front surface depth in screen space using a fast bilateral filter and (3) compose the final frame.

2.4.1 Computing Per-Pixel Fluid Data

In order to render fluids, our method needs the front and back depth of the fluid volume for each pixel of the view point. To this aim, we render each particle as a sphere. For the sake of performance, we do not render spheres using polygons but as point splats [148]. The back depth of the fluid volume is obtained using a reversed depth test. In order to obtain a connected fluid surface, point splats radius is 1.2 times larger than simulated particles radius. To accelerate this step, this buffer can be computed at a lower resolution than the screen. In our case, we used half the screen resolution.

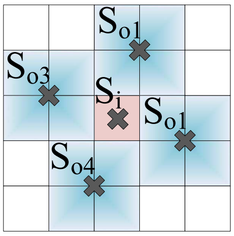


Figure 2.5 – The bilateral sampling kernel used to smooth the front surface of the fluid.

Using the raw depth values for the front fluid surface makes it appear blobby. To smooth out depth values, we could have used a bilateral Gaussian filter but this is a non separable blur filter. The algorithm proposed in [148] is very efficient but requires too many iterations to be effective. Thus, we propose to use a fast bilateral blur filter taking advantage of the hardware bilinear filtering (Figure 2.5). First, the depth is sampled for the inner pixel S_i. Then, four outer depths S_o are sampled. Thanks to hardware bilinear filtering, each of these outer samples represents the average of four depth values. The bilateral filtering is achieved by weighting outer depth samples according to the depth difference to the depth sampled at S_i. An outer sample is used only if the four averaged depths are from the fluid surface. We give a weight of 2.0 for S_i and 4.0 for S_o samples. This method allows us to average 17 depth values using only 5 texture samples. This bilateral filter is of lower quality than the filter proposed in [148] and can result in plateaus of equal depth. However, we can obtain a smooth fluid surface in few iterations. After preliminary testings we set the number of iterations to four.

2.4.2 Fluid Compositing

The fluid volume is finally composed with the VE during a single full-screen pass using the scene color S_c and linear depth S_d buffers, as well as the fluid data buffer. As in [148], the distortion of the scene color S_c perceived through the fluid is approximated as a texture look-up in screen-space with a displacement vector corresponding to the normal of the front fluid interface scaled by the fluid depth and refraction intensity. The final pixel color $col(p)$ is given by:

$$col(p) = \texttt{lerp}(\mathbf{F_c}, \mathbf{S_c}(\mathbf{p} + \beta \times T(p) \times \mathbf{n}.xy), \exp^{-T(p)*\mathbf{F_e}}) + \mathbf{k_s}(\mathbf{n} \bullet \mathbf{h})^{\alpha} \tag{2.12}$$

Fluid thickness $T(p)$ is first computed for each pixel p using

$$T(p) = \min(F_b D(p), S_d(p)) - \min(F_f D(p), S_d(p)) \tag{2.13}$$

where $F_f D$ and $F_b D$ are respectively the front and back depth of the fluid volume. In the case $T(p) = 0$, we simply copy the scene color. Otherwise, if $T(p) > 0$, we compute the final pixel color using Equation 2.12 where $\mathbf{F_c}$ is the fluid color, $\mathbf{F_e}$ is the wavelength-dependent color extinction coefficients of the fluid, β represents the amount of background color distortion simulating a refractive index, $\mathbf{h}$ the Blinn half-angle vector, $\mathbf{k_s}$ the specular color and α the specular exponent. The normal n is computed using finite difference on the front fluid surface depth [148].

The proposed optimizations of the technique described in [148] makes the approach suitable for real-time applications. These optimizations can thus be considered as a trade-off between performance and quality.

2.5 Evaluation

In this section, we evaluate our approach in terms of computation time and haptic rendering (forces and torques). The quality of our real-time visual rendering is also discussed.

2.5.1 Hardware Setup

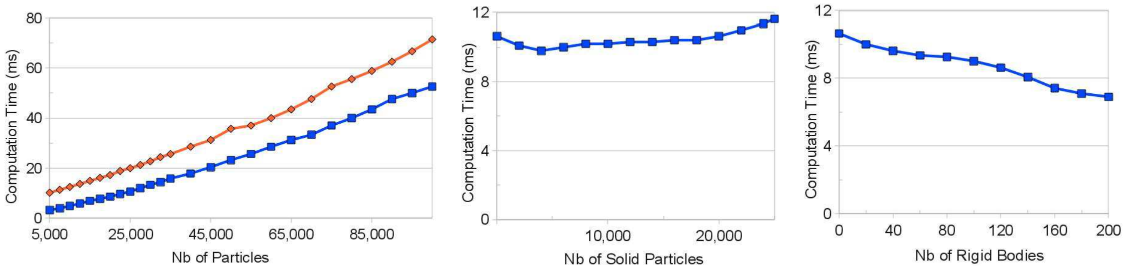
The evaluation scenarios were carried out using two Virtuose 6DoF force-feedback devices from Haption (Soulge-sur-Ouette, France). Figure 2.12 shows both devices in use. The simulations were run on a laptop computer with a Core 2 Extreme X7900 processor at 2.8GHz, 4GB of RAM memory, and a Nvidia Quadro FX 3600M GPU with 512MB of graphic memory.

2.5.2 Computation Time

2.5.2.1 Fluid Simulation Performance

Ideally, for rigid body simulations, update rates should be close to 1000Hz [50]. However, forces due to fluids do not change rapidly. Hence, in order to achieve a smooth and stable haptic rendering, the update rate can be considerably lowered, and rates close to 70Hz have been reported to be satisfactory [27] [26].

Figure 2.6a shows the performance of our fluid simulation with respect to the number of fluid particles. We measured the computation time of the simulation of a 1x1m fluid pool with and without the graphic rendering.



(a) (b) (c)

Figure 2.6 – Performance evaluation of our simulation algorithms. (a) Performance of the fluid simulation (in blue, squares) and the fluid simulation including graphic rendering (in red, diamonds) with respect to the number of fluid particles. (b) Performance of the simulation of a pool of fluid and a rigid body sphere with respect to the size of the rigid body. The number of total particles is constant. (c) Performance of the simulation of rigid bodies with constant total number of particles with respect to the number of rigid bodies.

Under 50ms (20 Hz), we consider that the simulation is not adequate for real time applications. With around 2,300 particles, the simulation and the graphic rendering take the same amount of time, while with 100,000 particles the simulation is around 4 times slower. Hence, the bottleneck of our simulation, when increasing the number of particles, is the physical simulation, while the graphic rendering remains efficient with a much lower drop in frame rate.

At 70 Hz, our implementation can thus simulate 32,000 fluid particles with the aforementioned hardware configuration, thus allowing the interaction with a considerable volume of fluid.

2.5.2.2 Unified Particle Model Performance

Figure 2.6b shows the computation time required to simulate a pool of fluid and a rigid body sphere with respect to the size of the rigid body. As the size of the rigid body increases, the size of the pool decreases to maintain a constant number of particles. The total number of particles is set to 25,000 since it allows a good frame rate for haptic rendering while being enough to simulate detailed objects and large amounts of fluid.

Up to 5,000 rigid body particles, the computation time decreases, due to the skipping of the density computation for rigid body particles. Beyond 5,000 particles, the computation time starts increasing again, since updating the rigid body particle positions from the newly computed values for the center of mass becomes more time consuming. In the GPU, each rigid body is treated in a single CUDA block in order to share the common rigid body values. When dealing with several rigid bodies of smaller size, which is more common than a VE with a single rigid body, the simulation becomes much more efficient, as shown in Figure 2.6c. From the single 25,000 particles rigid body sphere, we made several non-contacting rigid bodies spheres of equal size. When increasing the number of rigid bodies, with a constant number of total particles, we can see how computation time decreases due to the use of more CUDA blocks with a lower number of threads, making the particle update more parallel and efficient. Overall, the overhead due to rigid body simulation remains small, since in the worst-case scenario of a single rigid body the simulation takes less than 10% more time to compute than a pool of fluid with the same number of particles.

2.5.3 Graphic Rendering

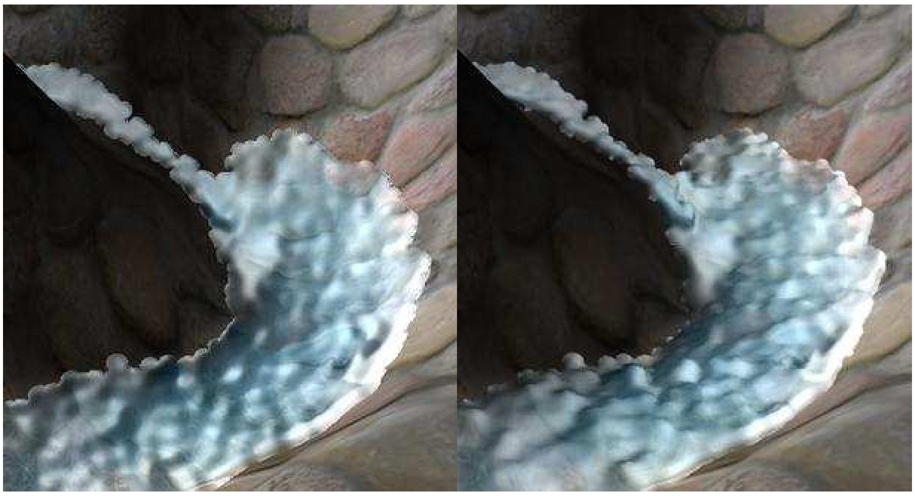


Figure 2.7 – Comparison of our graphic rendering method (left) with the original version [148] (right) for a volume of fluid.

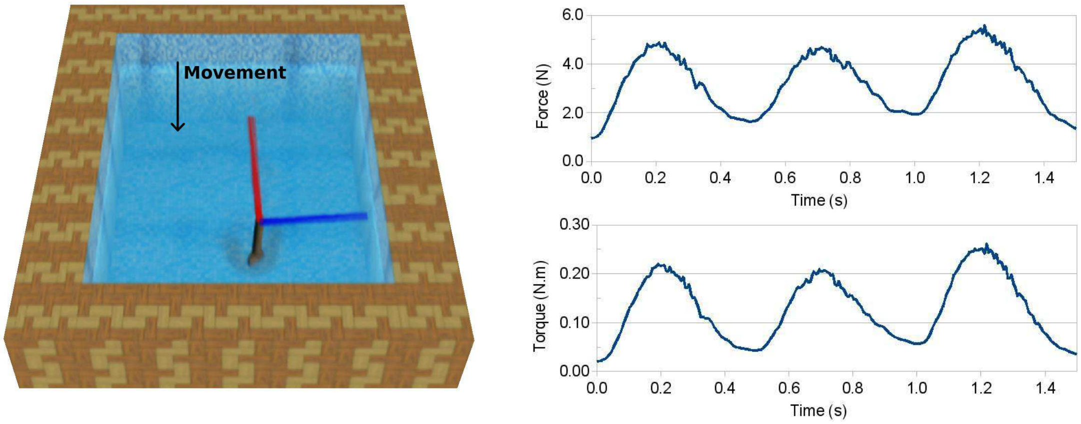
Figure 2.7 shows the result of our technique (left) and of the technique proposed in [148] (right) for a fluid volume made of 32,768 particles, rendered at a resolution of 1024x768. The method described in [148] requires 60 passes (GPU computation time of 45ms for a final frame rate of 20fps) to result in a smooth fluid surface. However, our method only requires 4 passes (GPU computation time of 8ms for a final frame rate of 90fps). Our method being designed primarily for speed over accuracy, some artifacts are visible at the edges of the volume whereas, using the original approach, the edges are preserved. In addition, our fluid rendering exhibits plateaus of equal depth producing a surface that seems more flat. However, our method is faster than the original method [148] and accurate enough for our use-cases.

2.5.4 Example Scenarios

Throughout the following examples, we highlight some interesting interaction scenarios to illustrate the various possibilities of our approach. We recorded the efforts exerted on the haptic device, and provide plots with the magnitude of the efforts with respect to time, in order to have a visual feedback on the haptic rendering.

2.5.4.1 6DoF Interaction

This scenario was designed to highlight the 6DoF capabilities of our technique. The possibility to perceive torques is a main feature of our approach, since these torques were very limited or not possible at all in prior existing work. The user exerts a forward and backward movement with a rigid body spoon on a pool of low viscosity fluid. Figure 2.8a shows the forces and torques involved in the stirring of the fluid with the spoon. We can notice the force (in red) going opposite to the back-to-front movement of the spoon. The torque (in blue) is generated through a rotation around an axis orthogonal to the linear movement. These visual cues match the plotting of the force and the torque over the duration of the movement (3 forward and backward movements) in Figure 2.8b. The user feels the force exerted on the spoon, as well as the corresponding torque, which plays a major role in the haptic perception of the stirring movement.



(a) (b)

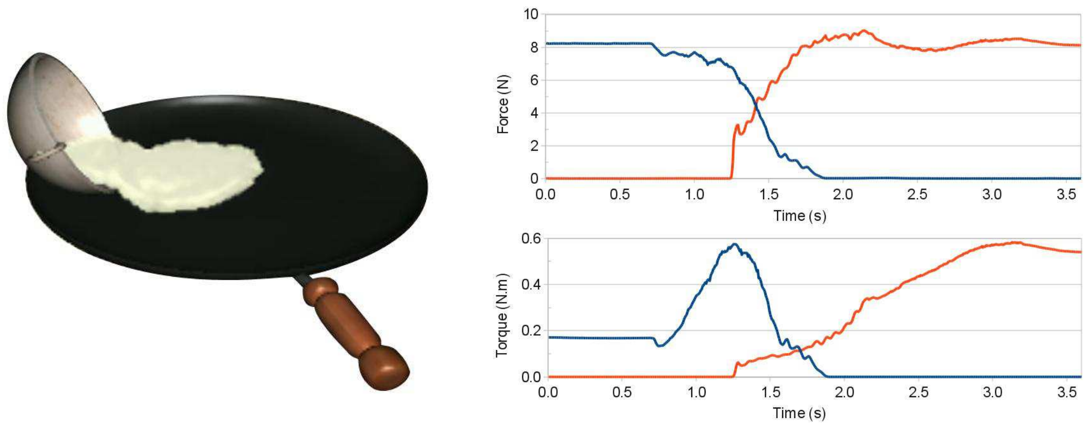
Figure 2.8 – 6DoF haptic interaction: a virtual spoon is used to stir a pool of fluid. (a) Force (in red) and torque (in blue) exerted on the spoon. The force is opposite to the spoon movement. The torque shows a clock-wise effort in accordance to the spoon movement. (b) Plot of the force (top) and the torque (bottom) generated by the forward, backward and forward movements of the spoon on the fluid.

2.5.4.2 Container Interaction

In this scenario, we illustrate the use of rigid bodies as containers to interact with the fluid. We used two 6DoF Virtuose haptic devices. The first device (left hand) is coupled to a bowl, while the second device (right hand) is coupled to a pan. The fluid is poured from the bowl into the pan, as shown in Figure 2.9a. Figure 2.9b shows the forces and torques involved in the pouring movement of the bowl. The fluid mass shift is clearly visible in the force plot, while the torque plot shows how the bowl is first tilted (torque peak) and then emptied (torque decrease).

2.5.4.3 Variable Viscosity

An important feature of our approach is the possibility to interact with fluids of different viscosity, and hence “feel” these different viscosities through the haptic devices. We



(a) **(b)**

Figure 2.9 – Container interaction: a bowl and a pan, each coupled to a 6DoF haptic device, are used as containers. (a) Fluid is being poured from the bowl into the pan, making the fluid mass shift between both rigid body containers. (b) Plot of the force (top) and the torque (bottom) of the bowl (in blue/dark) and the pan (in red/light) during the pouring motion of some fluid.

designed a scenario to capture the forces exerted on a rigid body container being spun around its vertical axis. The container is full of fluid, and initially at rest. The glass is spun with constant speed, kept rotating, and then stopped suddenly. The fluid has a viscosity of 2 Pa.s, while four different viscosities are used for the rigid body container: 2, 10, 20, 40 Pa.s. The increase in viscosity leads to higher forces exerted on the haptic device. Figure 2.10 shows the torques generated in each case, clearly showing each of the 4 phases (rest, start, spin, stop) of the movement and each of the 4 viscosities (from low to high: blue, red, yellow, green).

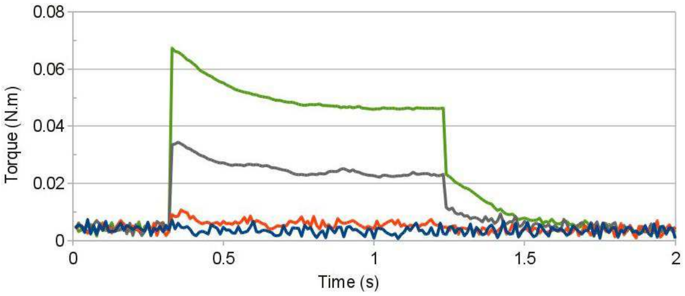


Figure 2.10 – Plot of the torques exerted on a rigid body container full of fluid (2 Pa.s viscosity) during a spinning movement. The container has different viscosities: 2 Pa.s (blue), 10 Pa.s (red), 20 Pa.s (yellow), 40 Pa.s (green).

2.5.4.4 Bimanual coupling on the same rigid body

Our haptic coupling and rendering mechanism allows the 6DoF haptic coupling between one or multiple haptic device and any rigid body, as well as between N devices and the same rigid body. This last feature of our approach is illustrated through a bucket full of low viscosity fluid held with two hands. The virtual bucket is coupled to the two 6DoF Virtuose haptic devices, as holding the bucket by each of its handles. The bucket is tilted to the left, to the right, and then to the left again, each time spilling some fluid. Figure 2.11 shows the force and torque plot for each haptic device during the tilting movements of the bucket. The peaks and valleys correspond to the tilting movements, increasing the force and the torque on the side to which the bucket is tilted. Forces and torques progressively decrease as fluid is spilled.

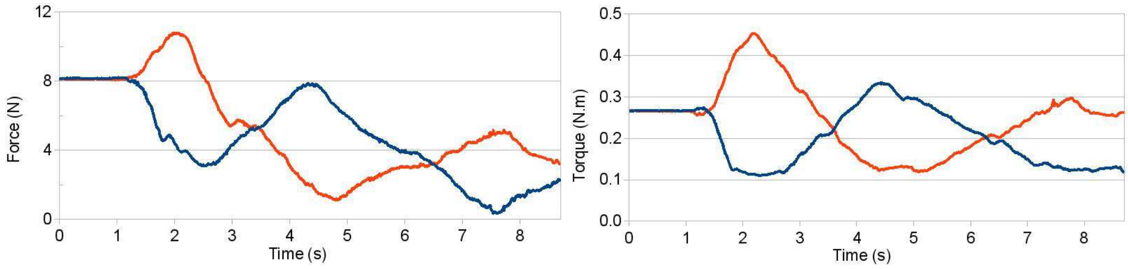


Figure 2.11 – Bimanual coupling on the same rigid body. Plot of the force (left) and the torque (right) generated by fluid inside a bucket coupled to two 6DoF haptic devices (left hand in red/light, right hand in blue/dark).

2.5.5 A complete use-case

To showcase the main features of our approach, we developed a complete use-case: the Virtual Crepe Factory, a virtual cooking simulator. It is a training and entertaining 2-handed interactive application that simulates the preparation process of a crepe. The user holds two virtual objects, a bowl and a pan, through two 6DoF Virtuose haptic devices from Haption, as shown in Figure 2.12. The simulation guides the user through all the steps required to prepare a crepe (Figure 2.13): from the stirring and pouring of the batter to the spreading of different toppings on top of the crepe. By preparing virtual crepes, users can experience 6DoF haptic interaction with fluids of varying viscosity. It is an example of the many promising applications that can be designed around 6DoF haptic interaction with fluids.

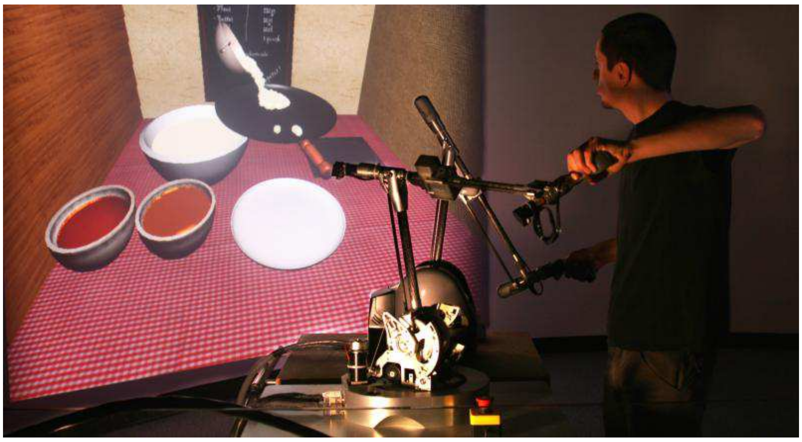


Figure 2.12 – The virtual crepe factory: the user pours some batter from the bowl and into the pan. He can feel the forces and torques from the pouring movement, as well as the weight shifting between his hands as the batter goes from the bowl to the pan.

2.6 Discussion

Table 2.1 compares our technique in terms of features with previous techniques allowing real-time haptic interaction with fluids, highlighting the important aspects of an interaction

Figure 2.13 – New interaction possibilities: a bowl coupled to the 6DoF haptic device is used to stir the batter (left), and the same bowl pours maple syrup on the crepe (right).

Method	Physical model	DOF	Real time	Perform. (Hz)[1]	GPU accel.	Arbitrary rigid bodies	Variable viscosity
[107]	Eulerian	3	✓	30			
[25]	-	6		500			
[27]	Eulerian	6	✓	40 - 70			
[26]	Eulerian	3	✓	30 - 75	✓		
Ours	Lagrangian	6	✓	60 - 120	✓	✓	✓

Table 2.1 – Comparison of our approach with previous work on real-time haptic interaction with fluids.

scenario. The following discussion is structured around these different features.

A main feature of our approach is the use of a Lagrangian SPH-based technique for haptic interaction with fluids, with many advantages in a real-time application: it is fast to compute (inherent mass conservation and no advection), it provides freedom regarding the VE (scalability, not bounded to a grid), and provides an intuitive way for haptic coupling. In the SPH model, the inherent smoothing of the SPH haptic forces, and the lack of discrete contact points, ensure that there are no discontinuities in the magnitudes of the resulting forces and torques, providing a stable and smooth haptic feedback.

Regarding the performance of our algorithms, we could run our scenarios at higher frame rates than other techniques, even with more complex VE, albeit in more recent hardware. The only exception is the work presented in [25], which is not entirely computed in real-time, limiting the interaction possibilities. The higher performance is achieved through the use of the SPH model and our efficient GPU implementation, and performing better than [149] and [132] (12 and 1.7 times faster, respectively), and close to [133] (1.2 times slower), which focus exclusively on the optimization of a particle-based simulation of fluids. The different force and torque plots of section 2.5.4 provide a visual representation of the force feedback. The haptic forces and torques exerted on the device match the scenarios and the different interactions that are visually perceived, showing that the haptic rendering is faithful and realistic.

The disadvantages of our technique are common to distance-field-based haptic rendering techniques. The use of a smoothing volume makes the interaction approximate, hence focusing the use of the technique on applications where exact surface representation is not required. As mentioned earlier, this approach is very well suited for the haptic interaction with fluids due to their approximate boundary.

---

[1] Regarding performance, fair comparison is difficult, since existing haptic fluid interaction techniques are based on different physical models and were simulated on different hardware. We provide these performance values for information purposes only.

Other approaches for solid-fluid interaction, such as direct forcing with constraint equations, provide more control on boundary conditions and enforce non-penetration for colliding particles. However, there is a computational cost with up to 3 collision detection steps for 2-way coupling scenarios, and some restrictions on the number of interacting bodies. Regarding the haptic coupling, the use of a Virtual Coupling mechanism improves the stability of the haptic feedback, but inevitably affects the transparency of the haptic coupling due to the introduction of a viscoelastic link between the haptic device and the rigid body. Further investigation will assess the impact of using a Virtual Coupling mechanism on the perception of fluid haptic feedback.

Although we did not run into stability issues, using explicit integration schemes means that unconditional stability is not guaranteed. In addition, if a particle moves more than a kernel-size in one time step it could pass through other particles without collision, no matter the solid-fluid interaction algorithm. However, this can only happen with very fast particles (above 10 $m.s^{-1}$ with our current simulation parameters), one order of magnitude faster than the highest speed of a fluid particle in our simulation scenarios. Further investigation is required to assess these issues, but they can only be solved with expensive implicit integration schemes, which would make computations overly expensive given the high number of particles and contact points.

Regarding its benefits, our haptic interaction technique provides real-time 6DoF force-feedback, which was only possible in [25] with precomputed ad-hoc forces. In addition, our unified particle model allows the seamless use of complex rigid bodies of arbitrary shape for the 6DoF haptic interaction with the fluid, including concave rigid body containers that can hold the fluid or parts of it. It allows an entire range of scenarios and applications, such as using complex utensils like a mixer to stir a soup, or odd shaped proxies to explore the cavities of human body vessels. Other advantages, not illustrated in this chapter but inherent to the SPH model, allow the design of rich and compelling VE: an entirely dynamic VE and topology-changing rigid bodies without impacting the computation time.

We believe many applications could be designed based on our approach. Such applications span from the medical field (organic fluids like blood) to industrial scenarios (painting, manipulating dangerous fluids) and gaming or entertainment simulations (water and mud in natural VE, water sports).

2.7 Conclusion

Fluids are widely present in industrial, medical and entertainment scenarios, as well as in our daily life. Allowing the haptic interaction with large volumes of fluid represents an important addition to VR. In this chapter, we proposed the first approach enabling 6DoF haptic interaction with viscous fluids.

Our novel approach is based on the Smoothed-Particle Hydrodynamics physical model. It allows real-time 6DoF haptic interaction with fluids of variable viscosity through arbitrary shaped rigid bodies. We proposed a novel haptic rendering technique, based on SPH forces. Thanks to a unified particle model, rigid bodies can interact with fluids and provide 6DoF haptic feedback. Using an efficient GPU implementation and a graphic rendering algorithm tailored to the system's requirements, our physical simulation reaches frame rates suitable for haptic interaction with large volumes of fluid. We designed different example scenarios to illustrate and evaluate some of the interaction possibilities offered by our technique. We showed how many rich and complex fluid manipulations are now easily achieved, with stable force and torque outputs. Interaction is no longer limited to 3 degrees of freedom, with torque feedback playing a major role in providing a compelling feeling of realism. Among these are the use of concave containers to hold fluid and hence transmit to the user force feedback coming, for example, from fluid stirring, pouring, shaking and scooping, as well as the inertia of the fluid inside the container.

This approach also introduces the Smoothed-Particle Hydrodynamics (SPH) model into the fluid haptic simulation, and, to the best of our knowledge, to the haptic realm in general, yielding positive results. Many issues of major concern for the haptic community are avoided in our approach through the use of the SPH model, such as the generation of smooth and stable haptic forces, the transparent handling of multiple contact points, the parallelizability of computations and their scalability, the seamless use of arbitrary-shaped objects, and the computation of a fully dynamic VE. By showing that haptic forces computed through the SPH model are particularly well suited for the interaction with fluids, haptic interaction with other media can now be imagined for the SPH model.

This novel approach proves that fluids are now readily available for complex 6DoF haptic interactions, and opens an entirely new horizon of applications and interaction techniques in VR. In addition, it is encouraging to see that SPH can be an adequate framework for generating haptic feedback from a physically based simulation. In the next chapter, we extend this approach to include the haptic interaction with deformable bodies, also simulated through SPH, thus providing a seamless way to interact with media in different states with force feedback.

Six Degrees-of-Freedom Haptic Interaction with the Different States of Matter

3

Contents

3.1 SPH Multisate Simulation **66**
  3.1.1 SPH Deformable Body Model 67
  3.1.2 Rigid Bodies, Fluids and Interaction Forces 68
  3.1.3 Changes of State 68
  3.1.4 Integration and Simulation Loop 69
3.2 6DoF Multistate Haptic Rendering **69**
  3.2.1 Rigid proxy 69
  3.2.2 Deformable proxy 71
  3.2.3 Proxy inducing state changes 71
  3.2.4 Friction Forces 71
3.3 Dual GPU Implementation **73**
3.4 Evaluation **74**
  3.4.1 Haptic Time Step 75
  3.4.2 Haptic Feedback 75
  3.4.3 Perceptual Evaluation 76
3.5 Discussion **78**
3.6 Conclusion **80**

Complex VE are not restricted to only one type of media. Generally speaking, complex VE allow the simulation of several types of media at the same time. In the graphics and simulation realm, media can be defined by three *states of matter*: rigid bodies, deformable bodies and fluids. Current physically based haptic techniques only focus on two states at most, thus limiting the interaction possibilities with complex VE. Furthermore, having the three states in the same simulation poses several challenges, in terms of computational demand and coupling mechanisms: specific haptic rendering approaches are used for each media, and specific interaction mechanisms are required between different media.

Recent work on physically based simulations has shown that the SPH framework is general enough to encompass the equations of continuum mechanics. However, update rates are usually not real-time, or fall short for haptic interaction. In [141], deformable bodies are modeled in real-time with SPH and the Moving Least Squares (MLS) algorithm to compute the elastic forces. The approach is improved in [150], allowing solid, deformable and fluid animation and interaction in a unified approach. A fully SPH-based approach is presented in [138], later corrected in [151] for a rotationally invariant formulation.

Thus, in this chapter, we introduce the first approach that provides physically based haptic feedback for fluid, deformable and rigid states of matter in the same simulation. We do so by extending our SPH haptic fluid interaction approach for the inclusion of deformable bodies alongside fluids and rigid bodies, thus providing a generic haptic coupling and rendering mechanism. Based on the SPH model for all three types of media, our method avoids the complexity of dealing with different algorithms and their coupling. We conducted a perceptual experiment to evaluate our approach by assessing the capability of users to recognize the different states of matter they interact with.

The chapter is organized as follows. Section 3.1 recalls the main equations involved in the SPH simulation of deformable bodies, and presents a simple mechanism to deal with state changes. Based on this multistate approach, the haptic coupling scheme is described in Section 3.2, enhanced with friction forces and allowing proxies of different state. A dual GPU implementation is described in Section 3.3, allowing to cope with the higher computational demands of continuum mechanics while maintaining haptic rates. The approach is evaluated in Section 3.4, in terms of performance and perceptual state recognition. We conclude after a discussion of our results in Section 3.5.

3.1 SPH Multisate Simulation

It is very well known that the three common states of matter are gas, liquid and solid. However, it is important to note that in the physical simulation realm a different subdivision is used, with the resulting states overlapping the formers. As shown in Figure 3.1, these simulated states of matter are fluids, deformable bodies and rigid bodies. A rigid body can be considered a deformable body with a high stiffness, and can be more efficiently approximated as an infinitely stiff body following its own dynamics. A gas is, to some extent, a fluid without viscosity. Hence, in the remaining of this chapter, we will refer to the states of matter as from a simulation point of view, namely fluids, deformable bodies and rigid bodies, which allow to simulate gas, liquid and solid physical states.

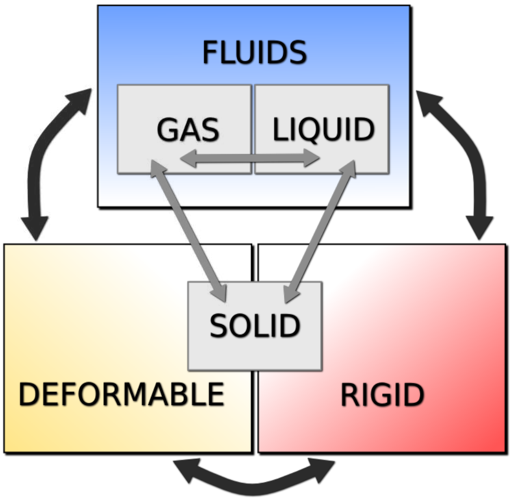


Figure 3.1 – The states of matter in a physical simulation: fluids, deformable and rigid bodies overlap the usual gas, liquid and solid states.

The SPH model can be efficiently used to simulate deformable bodies, alongside fluid and rigid body simulations presented in Chapter 2. In this section, we first recall the computational steps required to compute elastic forces as in [138], using the improvements proposed in [151] to make the computations rotationally invariant by using a corotational approach. For further details about the continuum mechanics approach, we refer the reader to Appendix A. We subsequently deal with the rigid and fluid states, the interaction forces between objects of different states, and the state change mechanisms. We then present the algorithm of the resulting simulation loop.

3.1.1 SPH Deformable Body Model

The rotation invariance has to be treated prior to computing the elastic forces, since the latter are computed from the locally rotated displacement field. For each particle i, the rotation matrix $\mathbf{R}_i$ based on its initial neighborhood is required. To this end, the matrix $\mathbf{A}_i$ is computed for each particle i:

$$\mathbf{A}_i = \sum_j m_j W(\mathbf{x}_i^0 - \mathbf{x}_j^0, h)\Big((\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j^0 - \mathbf{x}_i^0)^T\Big) \tag{3.1}$$

where $\mathbf{x}_i^0$ and $\mathbf{x}_j^0$ are the initial positions (in the undeformed state) of particles i and j respectively. $\mathbf{R}_i$ can then be extracted from $\mathbf{A}_i$ as its rotational part. A polar decomposition is therefore performed on $\mathbf{A}_i$, by first extracting the symmetric part $\mathbf{S}_i$ as:

$$\mathbf{S}_i = \sqrt{\mathbf{A}_i^T \mathbf{A}_i} \tag{3.2}$$

Then, $\mathbf{R}_i$ can be computed with:

$$\mathbf{R}_i = \mathbf{A}_i \mathbf{S}_i^{-1} \tag{3.3}$$

The elastic force exerted on a particle i is computed from the gradient of the locally rotated displacement field, $\nabla \bar{\mathbf{u}}_i$. Using an SPH formulation, this gradient is defined as:

$$\nabla \bar{\mathbf{u}}_i = \sum_j V_i^0 \bar{\mathbf{u}}_{ji} \nabla W(\mathbf{x}_i^0 - \mathbf{x}_j^0, h)^T \tag{3.4}$$

where $\bar{\mathbf{u}}_{ji}$ is the locally rotated displacement difference between neighboring particles j and i:

$$\bar{\mathbf{u}}_{ji} = \bar{\mathbf{u}}_j - \bar{\mathbf{u}}_i = \mathbf{R}_i^{-1}(\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{x}_j^0 - \mathbf{x}_i^0) \tag{3.5}$$

V_i^0 is the initial volume of particle i, a value that remains constant throughout the simulation and is computed using ρ_i^0, the density of particle i when the body is undeformed:

$$V_i^0 = \frac{m_i}{\rho_i^0} \tag{3.6}$$

From the gradient of the displacement field, the Jacobian $\mathbf{J} = \mathbf{I} + \nabla \bar{\mathbf{u}}_i^T$ can be computed. The Jacobian is then used to compute the strain ε through the Green-Saint-Venant strain tensor formula, which can cope with large deformations:

$$\varepsilon = \frac{1}{2}(\mathbf{J}^T \mathbf{J} - \mathbf{I}) \tag{3.7}$$

By considering Hookean and isotropic materials for our deformable bodies, the relation between stress σ and strain ε becomes linear, with $\sigma = \mathbf{C}\varepsilon$, and $\mathbf{C}$ is defined by only two parameters, the Young modulus and the Poisson ratio.

The elastic force $\mathbf{f}_{ji}$ exerted on a neighboring particle j by particles i is defined as the negative gradient of strain energy with respect to displacement, and can be computed as:

$$\mathbf{f}_{ji} = -2V_i^0(\mathbf{I} + \nabla\bar{\mathbf{u}}_i^T)\sigma_i\mathbf{d}_{ij} \tag{3.8}$$

with:

$$\mathbf{d}_{ij} = V_j^0\nabla W(\mathbf{x}_i^0 - \mathbf{x}_j^0, h) \tag{3.9}$$

Finally, the elastic force $\mathbf{f}_i^{elasticity}$ exerted on particle i is symmetrized and computed as:

$$\mathbf{f}_i^{elasticity} = \sum_j \frac{-\mathbf{R}_i\mathbf{f}_{ji} + \mathbf{R}_j\mathbf{f}_{ij}}{2} \tag{3.10}$$

3.1.2 Rigid Bodies, Fluids and Interaction Forces

The approach presented in Chapter 2 allowed us to interact with fluids and rigid bodies through the SPH framework. The addition of deformable bodies as a third physical model does not interfere with this previous approach: we use the same fluid and rigid body simulation algorithms.

One of the strengths of a unified approach is the use of unified interaction forces between objects of different media. This was the case in the approach presented in Chapter 2, using pressure and viscosity forces. When adding a third physical model, the unified interaction approach is further leveraged, improving the gain of parallel computation and providing a single mechanism for all possible interactions. Thus, we use the same unified interaction forces as in Chapter 2.

3.1.3 Changes of State

Changes of state can be easily added to a multiple state simulation [138]. In our approach, each particle carries a fluid state coefficient K^f and a deformable state coefficient K^d, both positive and with their sum varying between 0 and 1. These coefficients can be modified, producing changes of state. If $K^f = 1$ and $K^d = 0$, the particle behaves like a fluid. If K^f is lowered and K^d is raised, the particle starts moving to a deformable state. These changes are possible by multiplying fluid forces (pressure, viscosity) by the fluid state coefficient, and deformable body forces (elasticity) by the deformable state coefficient. Hence, all three forces are computed for each particle, and their magnitudes are scaled according to the aforementioned coefficients. However, in order to obtain symmetric forces, they are also scaled by the neighbor coefficients. Hence, the total force applied by particle j on particle i is:

$$\mathbf{f}_{ij}^{total} = K_i^f K_j^f(\mathbf{f}_{ij}^{pressure} + \mathbf{f}_{ij}^{viscosity}) + K_i^d K_j^d(\mathbf{f}_{ij}^{elasticity}) \tag{3.11}$$

A body *id* is carried by each particle, so that only particles with the same *id* can trigger the computation of internal forces. At any time, a particle belonging to a body can be set loose by changing its body *id*. For example, when the K^f of a deformable body particle reaches 1, its body *id* can be changed to make the particle leave the deformable body and join a pool of fluid nearby. Figure 3.11 illustrates a solidification mechanism, where some fluid batter is poured in a pan, then cooked into a deformable pancake, and then deposited on a plate.

3.1.4 Integration and Simulation Loop

The different models previously described are run simultaneously under the same SPH framework, improving the efficiency of the simulation compared to having three heterogeneous models. Hence, the simulation loop runs in four consecutive steps, which are detailed in Algorithm 3:

1. Compute corotational matrices $\mathbf{R}_i$ for each particle belonging to a deformable body (lines 1-3);
2. Compute new density and elastic properties (lines 4-9);
3. Compute new internal (between particles of the same entity) and interaction (between particles of different entities) forces (lines 10-16);
4. Integrate (compute new velocity and position) through a Leap-Frog integration scheme [131] (lines 17-30).

3.2 6DoF Multistate Haptic Rendering

In the previous section we showed how to simulate different states (fluid, deformable and rigid bodies) and their interactions with the same SPH model. In this section, we detail our haptic coupling scheme allowing to present the force feedback to the user through a multiple state proxy and a 6DoF haptic device. The user can therefore seamlessly interact with fluid, deformable and rigid media with 6DoF force feedback. Figure 3.2 illustrates this principle. Any virtual object of the VE can be either fluid, deformable or rigid, and the state is hidden inside the model. The interactions between the different states are also computed seamlessly based on the same model. In the end, a VE is populated by defining the state of each virtual object together with its corresponding physical properties (stiffness for deformable bodies, viscosity for fluids). Users can interact with the VE through a haptic device using the interaction forces, without ever dealing with the models behind the virtual objects.

3.2.1 Rigid proxy

For rigid body proxies, we can leverage the haptic coupling scheme presented in Chapter 2 for rigid-fluid haptic interaction. Since the scheme is based on the unified interaction forces, it can compute the force feedback from fluids, rigid bodies, and the newly added deformable bodies as well. We recall that, in this scheme, the rigid proxy is treated as a rigid body of the VE. When it interacts with different objects of the VE, no matter the state, the simulation computes the corresponding interaction forces as any other body from the VE. We use a specific integration step for the proxy, which includes forces and torques coming from the haptic device, as well as a Virtual Coupling mechanism [58] for an increase of stability [50].

The resulting 6DoF haptic coupling scheme is unified, in the sense that it allows the interaction with different media (fluid, deformable and solid) without distinction, since the haptic forces are computed in a unified way. This is confirmed by considering that, in the previous chapter, this scheme was used for fluid and rigid body interaction, and remains unchanged and performing efficiently when adding deformable bodies to the VE.

Algorithm 3 Physical simulation loop: a four-step algorithm

```
for all particles in deformable bodies do
    compute corotational matrices (Eqs. 3.1 to 3.3)
end for
for all particles in fluid and deformable bodies do
    compute density (Eq. 2.2)
    if particle in deformable body then
        compute displacement, strain and stress (Eqs. 3.4 to 3.7)
    end if
end for
for all particles in simulation do
    if particles from same entity then
        compute internal forces (Eqs. 2.5, 2.6 and 3.8 to 3.10)
    else
        compute interaction forces (Eqs. 2.5 and 2.6)
    end if
end for
for all particles in fluid and deformable bodies do
    add external forces
    integrate (new position and velocity)
end for
for all rigid bodies do
    for all particles in the rigid body do
        add force exerted on the particle to total force
        add torque from the particle to total torque
    end for
    add external forces and torques
    integrate (new rigid body position and velocity)
    for all particles in the rigid body do
        update particle position and velocity
    end for
end for
```

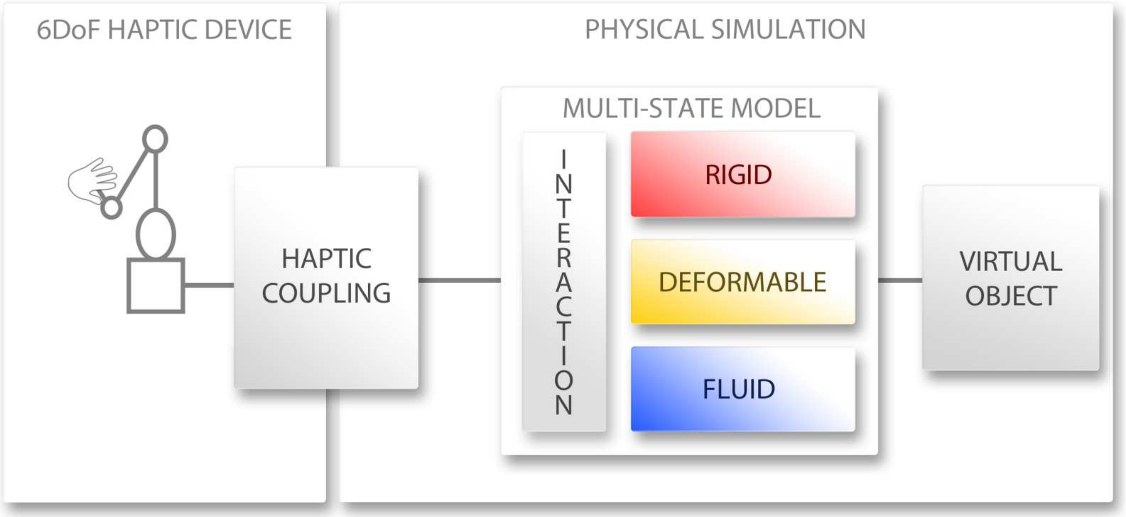


Figure 3.2 – 6DoF multistate haptic rendering: a user seamlessly interacts with a VE populated with virtual objects of different states of matter.

3.2.2 Deformable proxy

The use of a unified approach provides a great flexibility when dealing with bodies of different state of matter. The resulting unified haptic coupling scheme further leverages this approach by allowing haptic proxies of different states. We define a rigid core inside the proxy in order to have constant mass and inertia parameters, as well as a simple force and torque coupling mechanism with the haptic device. The rigid core is surrounded by media in the deformable state, and the internal elastic forces computed for the particles at the boundary of the rigid core become external forces exerted on the core. This simple approach has the advantage of requiring few changes in the coupling code, while providing a new proxy state as well as allowing dynamic changes in the proxy such as melting (section 3.1.3), plasticity and tearing.

3.2.3 Proxy inducing state changes

We can take advantage of our state change mechanism to seamlessly augment our coupling scheme with state changes induced by the proxy. This enables the design of more complex and compelling scenarios. An SPH temperature field, as described in [138], could be added around the haptic proxy rigid body. Heat emanating from the proxy is transfered to a deformable body, gradually changing the state coefficients of the particles in contact with the proxy. These particles move to a fluid state, until they melt completely and drop from the deformable body to form a fluid puddle on the floor. This procedure is transparent for the haptic coupling, and the user can perceive force feedback all along the process, interacting with the melting body and the resulting fluid.

3.2.4 Friction Forces

Friction arises from interactions between surfaces. Microscopic phenomena such as surface roughnesses and intermolecular attractions can modify the relative velocity of two bodies in contact. The Coulomb's law of friction has been widely used to model the friction force,

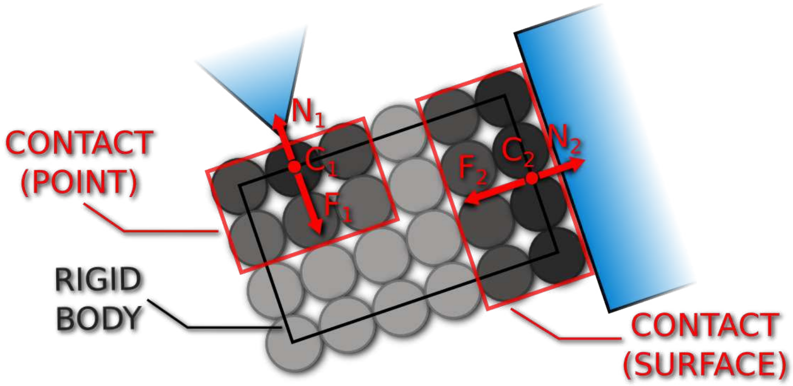


Figure 3.3 – Contact point and surface detection on a proxy rigid body.

since it provides a simple but accurate approximation to this phenomena at a macroscopic level. Since the friction force is highly dependent on the way surfaces contact each other, the correct computation of contact points and their corresponding contact forces is essential.

3.2.4.1 Contact point and surface detection

Due to the smoothed nature of SPH, a contact between two bodies involves more that only one contacting particle: in most cases, particles surrounding the contact point (or the contact surface) have non-zero interaction forces, as illustrated in Figure 3.3. These forces decrease with the distance to the contacting body, as intended by the smoothing kernel. Hence, computing the contact points and their corresponding interaction force requires taking into account the forces distributed among the neighboring particles.

Algorithm 4 details our approach for the detection of contact points within the rigid body proxy. Following the aforementioned reasons, we make two assumptions regarding contact force distributions. First, a contact is made of a particle (or a group of particles in the case of a contact surface) with an interaction force maxima and a set of neighboring particles with progressively lower forces. Second, the force decreases with the distance to the particle carrying the maxima. Hence, contact groups can be created as a set of neighboring particles surrounding a particle with a force maxima. Sorting the particles by force ensures that particles are treated only once: a particle without a neighbor in a contact group will start a new contact group as the particle with highest force within that group. In order to account for contact surfaces, neighboring groups with the same contact normal are merged together. Contact normals are computed by normalizing the interaction force vector of the group.

With this approach, a set of contact points and surfaces are extracted, each with a representative contact point C_p, a normal N_{C_p} and a force F_{C_p} (Figure 3.3).

3.2.4.2 Friction force computation

Once the contact points are extracted, we use Lazarevych *et al.* [152] approach for the computation of Coulomb friction forces. Static and dynamic cases friction cases can be detected by applying the maximal friction force to both bodies and then looking for a change of direction in the relative velocity of the bodies projected in the tangent space. The friction is dynamic if the direction has not changed, and static otherwise. The static

Algorithm 4 Detection of contact points and surfaces of a rigid body proxy

```
Discard proxy particles without interaction force
Sort remaining particles by decreasing interaction force
Set first particle as new contact group
for all sorted particles $p$ do
    if $p$ is a neighbor of existing contact group then
        add $p$ to contact group
    else
        create new contact group with $p$
    end if
end for
for all contact groups do
    compute contact force
    compute contact normal
end for
Merge neighboring groups with same contact normal
for all contact groups do
    set contact point $C_p$ at barycenter of particles with force maxima
    assign group normal $N_{C_p}$ and group force $F_{C_p}$ to $C_p$
end for
```

friction case is treated by setting a scaling factor on the friction force of each contact point, and then iteratively solving a non-linear system to globally ensure a zero relative velocity at each contact point.

Overall, our 6DoF haptic rendering technique provides a mechanism for the seamless interaction with media in rigid, deformable and fluid states, through rigid or deformable proxies, and taking into account friction forces during contact. However, an efficient implementation is required for high update rates, in order to generate stiff contacts. This implementation is described in the following section.

3.3 Dual GPU Implementation

The entire physical model is implemented on GPU using the CUDA framework [134]. It is based on our previous implementation, presented in Chapter 2, making extensive use of texture lookups, benefiting from texture data caching when retrieving the different data arrays.

In a haptic application, when running simulation computations on the GPU as well as visual rendering, both the haptic and the visual loops have a tight link, due to the concurrent use of the GPU. Since both loops are running on the same GPU, using respectively a CUDA and an OpenGL context, they cannot run in parallel, but only in sequence. Simulating multiple iterations of the haptic loop followed by one iteration on the visual rendering loop is not a viable compromise, since it would introduce gaps in the haptic update rate. The haptic update rate is therefore linked to the visual update rate, greatly undermining the haptic performance.

Taking advantage of the common availability of dual GPU setups in consumer workstations as well as laptop computers, we decoupled the haptic and graphic loops by using two GPUs, each controlled by its own thread running on its own CPU core. Figure 3.4

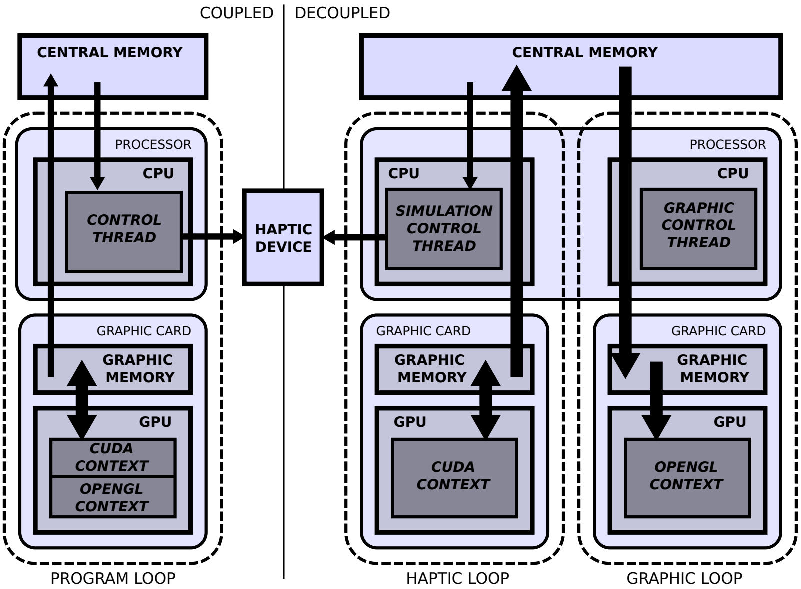


Figure 3.4 – Illustration of the decoupled architecture: while the coupled version (left) has a single loop, the new decoupled version allows two distinct and independent loops. The arrows represent data transfers.

illustrates this architecture, and compares it to a coupled, single GPU version.

In the new decoupled architecture, a simulation thread running on its own CPU core at the haptic rate f_h controls the CUDA computations running on the first GPU. The CUDA context reads and writes data to the graphic memory, corresponding to different arrays used and updated by the computations, which are position, velocity and force arrays. After each iteration, only the data required for visual rendering is copied to central memory, namely position and haptic coupling data. This coupling data is then sent from central memory to the corresponding haptic devices.

The graphic loop runs in parallel of the haptic loop, at the visual rate f_v independent of f_h. A graphic thread running on its own CPU core copies the position data from central memory to the graphic memory of the second GPU. It then controls the OpenGL rendering operations of the second GPU.

This decoupled implementation results in a gain of up to one order of magnitude for the haptic loop, as seen in Section 3.4.1. Although it is required to write the position data to central memory, this operation accounts for less than 1% of the haptic loop computation time. Only twelve bytes have to be copied per particle (3 position floats), which corresponds to as little as 352KB for 30,000 particles.

This performance boost is an essential improvement to obtain a realistic haptic feedback with rich and complex VE.

3.4 Evaluation

In this section we evaluate our approach, in terms of performance by measuring the update rate, and in terms of subjective appreciation by conducting a perceptual evaluation with users interacting with different media.

The evaluations were carried out using a Virtuose 6DoF force-feedback device from Haption Company (Soulge-sur-Ouette, France), and a computer with a Core 2 Extreme X7900 processor at 2.8GHz, 4GB of RAM memory, and two Nvidia GeForce 460 GT GPU

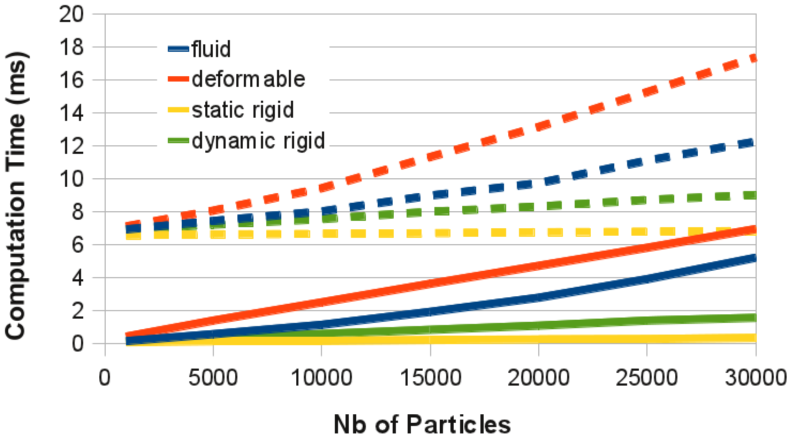


Figure 3.5 – Haptic time step in milliseconds for VE made of either a rigid body (static and dynamic), a deformable body or a fluid volume, according to the number of particles, with coupled (dashed curves) and decoupled (continuous curves) GPU implementations.

with 1GB of graphic memory.

3.4.1 Haptic Time Step

We measured the haptic time step of the simulation in order to evaluate the performance of our dual GPU implementation. We compare these results to a single GPU implementation. Four VE were used, each with either a fluid volume, a deformable body, a rigid body, and a static rigid body that can exert, but not receive, interaction forces. The VE have the same size and shape (a cuboid with a square base of 0.5m side length and a variable height according to the number of particles). We measured the time between two haptic updates (when data is sent to the haptic device), shown in Figure 3.5 in milliseconds, for a simulation ranging from 1,000 to 30,000 particles.

The performance improvement due to the decoupling of the haptic and rendering threads is clearly visible for all the states of matter. The haptic time step in the decoupled case corresponds to the haptic loop of Figure 3.4. For the coupled case, the haptic time step corresponds to the program loop, hence inevitably including the visual rendering. The visual rendering step takes around 10ms for the range of particles used in the measurements (for more details, we refer the reader to Chapter 2). Hence, there is an overall overhead of around 6ms in the coupled scheme, visible in the plot as a shift of the corresponding curves.

3.4.2 Haptic Feedback

We provide a visual representation of the forces exerted on a coupled rigid body interacting with the three types of media, in order to show the correspondence of the simulation behavior with the expected physical behavior.

A 20cm radius sphere, coupled to a 6DoF Virtuose, is dropped onto a 50x50x20cm block of particles, representing a volume of fluid, a deformable body or a rigid body. Figure 3.6 shows the forces involved in the interaction between the coupled sphere and the media. The three interaction patterns are computed and recorded individually, but are shown on the same graph for comparison purposes.

When the media is solid, the sphere bounces quickly, generating a force peak. When the media is deformable, the sphere compresses the media's surface until its elastic behavior pushes the sphere back into the air. When the media is fluid, the sphere penetrates the fluid until pressure makes the sphere rise to the surface and float.

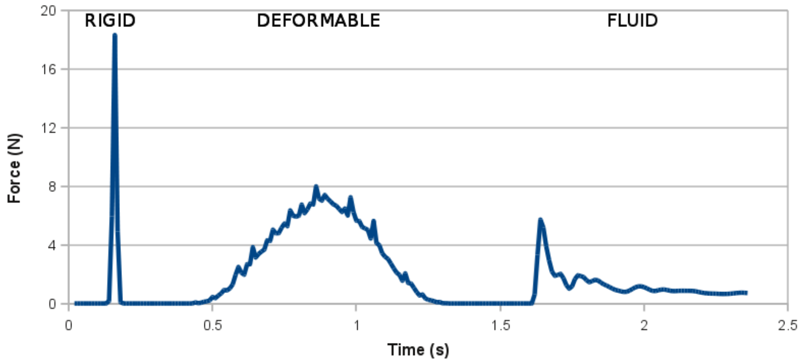


Figure 3.6 – Forces exerted on a sphere successively dropped on rigid, deformable and fluid media.

3.4.3 Perceptual Evaluation

In order to qualitatively evaluate our haptic multi-state approach, we conducted a perceptual experiment where subjects interacted with different types of media. Our goal was to assess the capacity of subjects to recognize the state of the matter present in the VE (fluid, deformable body, rigid body), in three different conditions: haptic feedback alone, visual feedback alone, and both at the same time. We recorded their answers on the state of matter they recognized. Their subjective appreciation was collected through a preference questionnaire.

3.4.3.1 Population

Twelve participants (2 females and 10 males) aged from 23 to 26 (mean = 24.8, sd = 0.7), took part in this experiment. None of them had known perception disorders. They were all naïve to the purpose of the experiment.

3.4.3.2 Experimental Apparatus

The experiment was conducted with a Virtuose 6DoF force-feedback device for haptic interaction, and a 17' flat screen in front of the subject for visual display. A carved cardboard box was used to situate a virtual tray containing the media in the VE. The simulation was computed using the aforementioned configuration. Figure 3.7 shows the experimental conditions. The VE consisted of a rigid body spoon coupled to the 6DoF haptic device positioned on top of a 60x60cm tray containing the media in either fluid, deformable or rigid state, as show in Figure 3.8. The media was five-particle-layers thick. The entire VE was rendered with a gray texture.

3.4.3.3 Procedure

Before the beginning of the experiment, the subject was given the instructions in written form. Before each trial, the subject had to position the handle of the haptic device at the same starting position, instructed beforehand. The participant could explore the VE

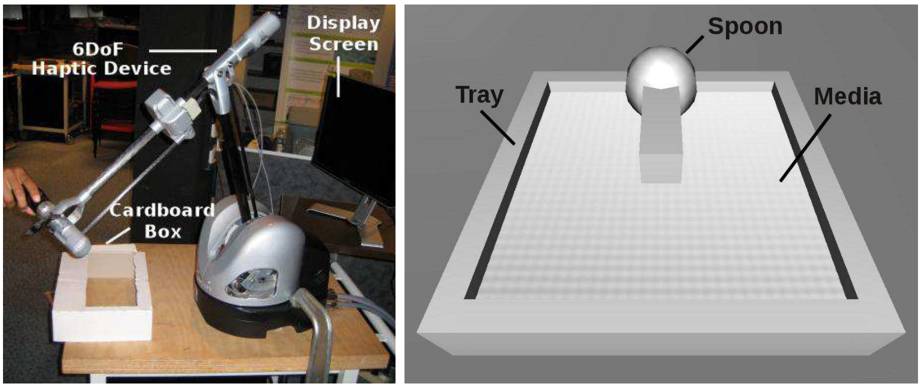


Figure 3.7 – Experimental apparatus: display devices (left) and VE (right) used in the experiment.

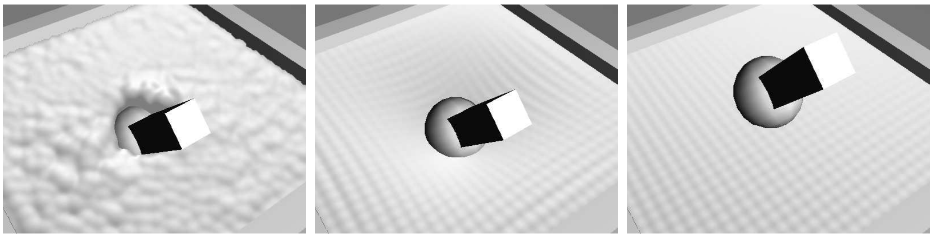


Figure 3.8 – The three different states of matter. From left to right: fluid, deformable, rigid.

freely up to the cardboard limits. The subject had 10 seconds to detect the state of the media he was interacting with before the feedback (visual and haptic) was turned off. He would then select the answer from the 3 possible choices on the screen. The experiment lasted about 30 minutes.

3.4.3.4 Experimental Plan

Participants completed all three feedback conditions (haptic, visual, and haptic+visual) and the order of the conditions was counterbalanced across participants. In each condition, the participants were exposed to 8 successive blocks of 3 trials corresponding to the three states, in random order. Participants completed a total of 72 trials (3 feedback conditions × 3 states of matter × 8 trials per state). For each trial and each subject, we recorded the answer given by the subject on the state of matter he recognized.

3.4.3.5 Results

We first evaluated the probability of correct answers for the different states of matter with one of the 3 conditions: haptic (H), visual (V) and haptic+visual (HV). It reveals that the probabilities of correct answers were the following: $p_H = 0.875$ for the haptic condition, $p_V = 0.99$ for the visual condition and $p_{HV} = 1$ for the haptic+visual condition. If we differentiate the different states of matter, we obtain the probabilities of correct answers shown in Table 3.1.

We performed likelihood ratio tests (LRT) to compare three statistical models: (1) $pV \neq pH$, $pV \neq pHV$, $pH \neq pHV$; (2) $pH \neq pV$, $pH \neq pHV$,$pV = pHV$; (3) $pV = pH = pHV$. LRT revealed that the second model is significantly more likely than the two other models. Thus, without taking into account the different states of matter (fluid, deformable or rigid), the probability of correct answers for H condition was significantly lower from

Table 3.1 – Probabilities of correct answers for each state of matter according to the condition: visual (V), haptic (H) and haptic+visual (HV).

	Deformable	Fluid	Rigid
V	1	1	0.99
H	0.87	0.86	0.89
HV	1	1	1

the ones for V and HV conditions (p-value< 0.001). The probability of correct answers for HV was not significantly different from the V condition (p-value$= 0.24$).

Considering only the deformable model, the probability of correct answers for H condition was significantly lower than the ones for V and HV conditions (p-value< 0.001). The probability of correct answers for HV was not significantly different from the V condition (p-value$= 1.0$). Considering only the fluid model, the probability of correct answers for H condition was significantly lower than the ones for V and HV conditions (p-value< 0.001). The probability of correct answers for HV was not significantly different from the V condition (p-value$= 1.0$). Considering only the rigid model, the probability of correct answers for H condition was significantly lower than the ones for V and HV conditions (p-value$<$ 0.001). The probability of correct answers for HV was not significantly different from the V condition (p-value$= 0.24$).

3.4.3.6 Subjective Questionnaire

After completing the experiment, a preference questionnaire was proposed in which participants had to grade from 1 to 7 the three feedback conditions according to 3 subjective criteria: (a) *degree of realism*, (b) *ease of use* and (c) *global appreciation*.

We performed an ANOVA on each of the subjective criteria. Concerning the realism criterion, it reveals a significant effect of the condition ($F(2, 33) = 10.11$, p-value< 0.001). A post-hoc analysis revealed that HV condition was significantly higher rated than V condition (adjusted p-value< 0.001) and H condition (adjusted p-value$= 0.005$). Concerning the ease of use, the ANOVA revealed a significant effect of the condition ($F(2, 33) = 11.39$, p-value< 0.001). A post-hoc analysis revealed that HV condition was significantly higher rated than H condition (adjusted p-value< 0.001). Concerning the global appreciation, we performed an ANOVA and we found a significant effect of the condition ($F(2, 33) = 9.1027$, p-value< 0.001). A post-hoc analysis revealed that HV condition was significantly higher rated than V condition (adjusted p-value$= 0.004$) and H condition (adjusted p-value$=$ 0.001). These results are highlighted in Figure 3.9.

3.5 Discussion

With a dual GPU implementation, our multi-state haptic rendering approach can reach high frequencies, even with a time consuming visual rendering. For a body made of 10,000 particles, update frequencies are close to or over 1000Hz for rigid bodies and fluids, and 400Hz for deformable bodies. It represents an improvement of 260% for deformable bodies, 700% for fluids, and above 1200% for rigid bodies. The performance gain is roughly proportional to the haptic loop frequency, if we consider a constant computation time for the visual rendering in the single GPU scheme. As the frequency increases, the performance gain compared to the single GPU scheme increases at a similar rate, and

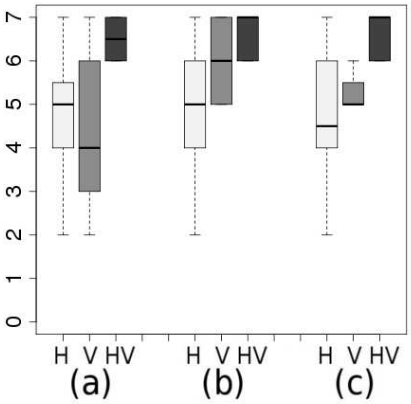


Figure 3.9 – Results for subjective ratings for the three criteria: (a) degree of realism, (b) ease of use and (c) global appreciation. Each boxplot is delimited by the quartile (25% quantile and 75% quantile) of the distribution of the condition over the individuals.

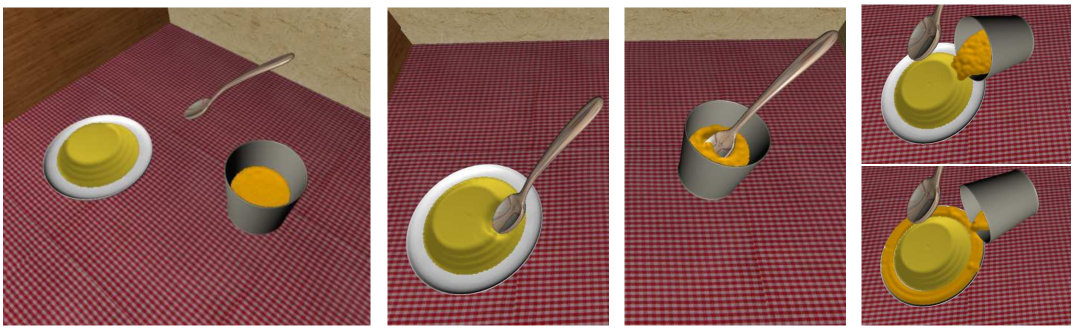


Figure 3.10 – Scenario illustrating a cooking simulator. The user, through a spoon coupled to the 6DoF haptic device, can touch and play with a custard tart, a glass and its liquid content. Haptic feedback is seamlessly computed for the three states of matter simultaneously.

vice-versa. This improved efficiency allows the design of complex VE while preserving the update rates required for high-quality haptic feedback.

Regarding the perceptual evaluation, the experiment described in Section 3.4.3 shows very high recognition rates for all three conditions (H, V, HV), and for all three states of matter (fluid, deformable, rigid). In the context of the experiment, our haptic multi-state approach succeeds in simulating the three different states of matter, and in providing haptic feedback to the user with a high degree of realism.

Results for V and HV show that recognizing the state of matter with visual feedback was a rather straightforward task, with almost perfect scores for both conditions, due to the choice of very representative conditions for each state of matter. This results suggests a realistic simulation in terms of physical model. However, the strongest result of this experiment is the recognition rate for the Haptic only condition (H), which is above 87% in average, showing a highly realistic haptic feedback.

We did not find significant results in terms of state recognition between both conditions with visual feedback since, again, visual cues seem to be enough to convey the state of matter to the users. This was also highlighted by many subjects when filling the questionnaire. Hence, the addition of haptic feedback did not significantly contribute to the recognition of the type of media they were interacting with, when the visual component was present. However, when referring to the results of the subjective questionnaire, it is

Figure 3.11 – Scenario illustrating the changes of state. The user can interact with fluid, deformable and rigid states through a cooking simulator. Some fluid batter held by a rigid cup is poured into a rigid pan (left), then progressively cooked into a deformable body (center), and then deposited on a solid plate (right).

clear that the addition of haptic feedback on top of visual rendering increased both the degree of realism and the general appreciation of the subjects. Their feedback was very positive regarding the combination of both modalities, and is in accordance with previous studies evaluating the combination of haptics with other modalities [111, 112, 113]

Taken together, our approach efficiently fulfills its objectives, by allowing the simultaneous interaction with different states of matter in the same VE with haptic feedback, and improving the interaction experience. Potential applications span from medical training to industrial and entertainment simulations. In this scope, we designed a virtual kitchen scenario with familiar objects very representative of their type: a solid glass full of thick orange juice, a wobbly custard dessert and a spoon coupled to the 6DoF haptic device. Users can freely explore the VE through the spoon and better perceive each object through the sense of touch. Figure 3.10 shows how a user interacts with the different media present on the VE, while everything follows a physically based behavior. The state change mechanism is illustrated in Figure 3.11. Some fluid batter is poured in a pan, then cooked into a deformable pancake, and then deposited on a plate.

3.6 Conclusion

The simultaneous haptic interaction with media in different states poses several challenges in terms of computational demand and coupling mechanisms. In this chapter, we tackled this problem by introducing the first multistate haptic rendering approach, allowing haptic interaction simultaneously with media in fluid, deformable and rigid states.

We extended our SPH haptic fluid interaction approach for the inclusion of deformable bodies alongside fluids and rigid bodies, by bringing on Solenthaler *et al.* [138] and Becker *et al.* [151] work to the required speeds. Using the SPH physical model for all three states of matter, our method avoids the complexity of dealing with different algorithms and their coupling. Our haptic coupling mechanism is also extended to encompass deformable bodies, thus providing a unified haptic coupling mechanism to render interaction forces and torques from a multistate VE through a 6DoF haptic device. We show how the proxy object, coupled to the haptic device, can be of different states, while inducing and undergoing state changes. Friction forces can also be computed through this approach. We achieve high update rates thanks to a dual GPU implementation, using separate GPUs for haptic and visual rendering tasks, thus reaching the frequency thresholds for a good haptic perception.

We evaluated our technique by conducting a perceptual experiment. Our objective was to assess the capability of users to recognize the different states of matter they interacted with. Results showed a high recognition rate, even when providing only haptic feedback, and the increased appreciation of users when combining haptic and visual cues.

We illustrated our technique with virtual cooking examples showing potential VR applications.

Vibrotactile Rendering of Fluids

4

Contents

4.1 Overview 84
4.2 Previous Approaches for Real-Time Fluid Sound Synthesis 84
4.3 Enhancing a Real-time Fluid Simulation with Bubbles 85
4.4 Vibrotactile Model 86
  4.4.1 Initial Impact 86
  4.4.2 Harmonic Bubbles 87
  4.4.3 Main Cavity Oscillation 88
4.5 Vibrotactile Rendering 89
4.6 Extension to Other Modalities 90
4.7 User Feedback 91
  4.7.1 Scenario 91
  4.7.2 Discussion 91
4.8 Conclusion 92

Although a significant addition to a VR system, force and visual feedback are not the only modalities to convey important cues. Vibrotactile and acoustic feedback can provide crucial information to the user when interacting with a VE. Vibrotactile feedback can generate higher-frequency components that are not captured by the kinesthetic modality, providing, for instance, texture and transient cues. Acoustic feedback allows even wider frequency ranges, conveying information about material type, impact magnitude and distance. Both modalities can be displayed through cheap and off-the-shelf devices, making them a very attractive addition to VR setups.

Some common materials with which we interact on a daily basis can be simulated and displayed through the vibrotactile modality, such as wood, metal [59, 30] and aggregates like gravel and snow [31], leading to compelling multimodal VR scenarios [31, 32]. However, fluids have again been largely ignored in this context. For VR simulations of real world environments, the inability to include interaction with fluids is a significant limitation. The approach described in this chapter represents an initial effort to remedy this, motivated by our interest in supporting multimodal VR simulations such as walking through puddles or splashing on the beach.

The main contribution of this chapter is the introduction of the first physically based vibrotactile fluid rendering model for solid-fluid interaction. Similar to other rendering approaches for virtual materials [31, 32, 30], we leverage the fact that vibrotactile and acoustic phenomena share a common physical source. Hence, we base the design of our vibrotactile model on prior knowledge of fluid sound rendering. Using this approach, we are exploring the use of bubble-based vibrations to convey fluid interaction sensations to users. We render the vibrotactile feedback for hand-based and, in a more innovative way, for foot-based interaction, engendering a rich perceptual experience of feeling the sensations of water.

The chapter is organized as follows. Section 4.2 reviews existing algorithms for the real-time synthesis of fluid sound. Inspired from these approaches, and since fluid sound is generated mainly through bubble and air cavity resonance, we develop an efficient bubble generation mechanism in Section 4.3, on top of our fluid simulation. We then introduce our novel vibrotactile model in Section 4.4, enabling rich body-fluid interactions with vibrotactile feedback, presented in Section 4.5, and multimodal feedback, presented in Section 4.6. We collected user feedback through a pilot study described in Section 4.7, showing positive and encouraging results.

4.1 Overview

When an object vibrates under an applied force, a pressure wave is generated at its surface, traveling to the subject's ears and mechanoreceptors. We motivate our vibrotactile approach, based on sound generation mechanisms, on the fact that acoustic and tactile feedback are both vibrations that share a common physical source.

By comparing film frames with the air-borne generated sound, Richardson [153] provides an explanation for the process of a projectile impacting and entering a fluid volume. The impact produces a "slap" and projects droplets, while the object penetration creates a cavity that is filled with air. The cavity is then sealed at the surface, creating an air bubble that vibrates due to pressure changes. Smaller bubbles can spawn from the fragmentation of the main cavity, as well as from the movement of the fluid-air interface, such as when the droplets return to the fluid volume.

Our vibrotactile model is therefore divided in three components, following the physical processes that generate sound during solid-fluid interaction [153, 154]: (1) the initial high frequency impact, (2) the small bubble harmonics, and (3) the main cavity oscillation. As a consequence, it is highly dependent on the efficient generation and simulation of air bubbles within the fluid. Hence, a real-time fluid simulator enhanced with bubble synthesis is required on the physical simulation side.

Figure 4.1 provides an overview of our approach. The physical simulator automatically detects the solid-fluid impacts and the creation of air bubbles caused by interaction between a solid (such as a foot, a hand or an object) and the fluid volume. For each of these events, it sends the corresponding message to the vibrotactile model, which synthesizes a vibrotactile signal according to the simulation parameters. The signal is then output through a specific vibrotactile device, such as an actuated tile for foot-fluid interaction, or a hand-held vibrator for hand-fluid interaction.

4.2 Previous Approaches for Real-Time Fluid Sound Synthesis

In this section, we overview previous work on real-time fluid sound synthesis. This survey is not available in Chapter 1 since it is not directly related to haptics.

As many approaches for the vibrotactile rendering of solid materials [31, 32, 17, 30], we aim at leveraging real-time fluid sound synthesis algorithms to generate the corresponding vibrotactile feedback. Those techniques that are physically based rely on the oscillation of air bubbles trapped inside the fluid volume [153] to produce sound. The first bubble sound synthesis technique was proposed in Van den Doel's seminal work [155] where, based on

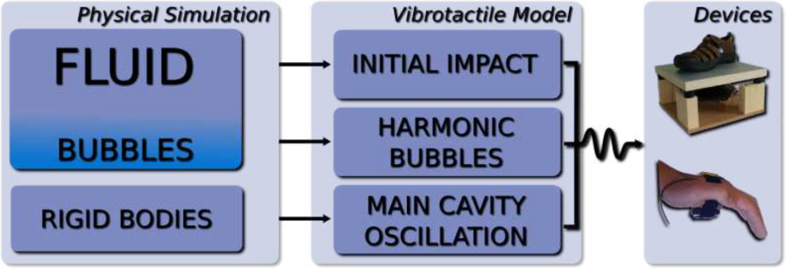


Figure 4.1 – Overview of our approach: the physical simulation computes the different parameters that are fed to the 3-step vibrotactile model, producing the signal sent to the various vibrotactile displays.

Minnaert formula [156], he provides a simple algorithm to synthesize bubble sounds based on a few parameters. However, the synthesis was not coupled to a fluid simulation. This is achieved by Drioli *et al.* [157] through an ad-hoc model for the filling of a glass of water, based on the height of the fluid inside the glass and on collision events. Moss *et al.* [158] propose a simplified, physically inspired model for bubble creation, designed specifically for real-time applications. It uses the fluid surface curvature and velocity as parameters for bubble creation and a stochastic model for bubble sound synthesis based on Van den Doel's work [155]. However, the model is designed for a shallow water simulator, which greatly reduces interaction possibilities by allowing only surface waves, precluding splashes and object penetration.

4.3 Enhancing a Real-time Fluid Simulation with Bubbles

The starting point of this approach in our fluid and rigid body simulation presented in Chapter 2. We use the SPH model for both fluids and rigid bodies, and a full GPU implementation for real-time speeds.

As previously mentioned, in order to achieve vibrotactile interaction with fluids we need to simulate the bubbles inside the fluid. Since we only seek bubble creation events resulting in bubble sound synthesis, a bubble has a very short life span within our model, and can be seen more as an event than as the actual simulation of a pocket of air. Hence, we adopt and simplify an existing SPH bubble simulation algorithm [143] to obtain an efficient bubble creation and deletion mechanism.

A bubble is spawn when a volume of fluid entraps a volume of air. In order to detect this phenomenon within the SPH simulation, we compute an implicit color field c^p as in the method of Muller *et al.* [143]. This color field estimates the amount of neighboring particles (fluid, rigid and bubble) around any position in space, while its gradient ∇c^p estimates in which direction the surrounding particles are mainly located. At each time step, we compute ∇c^p at each fluid particle position with:

$$\nabla c_i^p(\mathbf{x}_i) = \sum_j V_j \nabla W(\mathbf{x}_i - \mathbf{x}_j, h) \tag{4.1}$$

A fluid particle i triggers a bubble creation if the following conditions are fulfilled:

- the vertical component of ∇c_i^p is positive: the fluid particle has most of its surrounding particles above it, creating a pocket of air under it.
- the magnitude of the velocity of the particle is above a threshold: still or slow moving fluid particles do not generate bubbles.

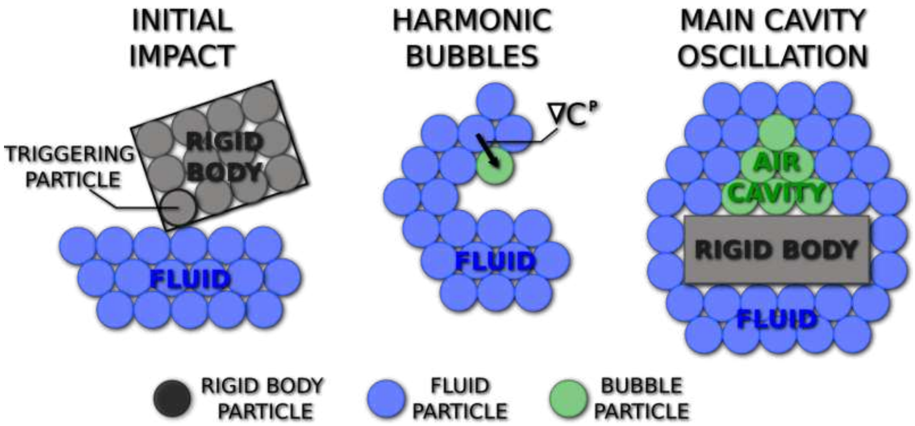


Figure 4.2 – The three components of our vibrotactile model.

A bubble is destroyed when it is not entrapped by fluid or held by a rigid body anymore. Since we only use bubbles for triggering events, a bubble is also destroyed if it is alone in the surrounding media. To this end, we compute another implicit color field, c^b, which only considers bubble particles, thus estimating the amount of neighboring bubbles surrounding a point in space. A bubble i is destroyed if one of the following conditions is fulfilled:

- the vertical component of ∇c_i^p is negative: the bubble particle has most of its surrounding particles under it, and the air cannot be trapped anymore.
- the color field c_i^b is null: the particle is alone inside the media.

4.4 Vibrotactile Model

Our vibrotactile model receives the events from the physical simulation, and can synthesize a signal through 3 different components: the initial high frequency impact, the small bubble harmonics, and the main cavity oscillation, as illustrated in Figure 4.2.

4.4.1 Initial Impact

During rigid body impacts on a fluid surface, Richardson [153] observed a damped high-frequency and low amplitude sound immediately after the impact, later explained as a guided acoustic shock [159]. To the best of our knowledge, no model provides the equations for air pressure oscillations due to a rigid body impact on a fluid surface. Previous work has been able to model the phenomenon to some extent, only for very simple shapes and specific penetration cases [160]. Nevertheless, the short duration of the impact does not justify a computationally expensive implementation. Hence, similar to previous work [157], we follow a physically inspired approach exploiting the short and burst-like nature of the vibration.

4.4.1.1 Synthesis

The impact signal is synthesized in a three step approach. A burst of white noise is first generated, spanning on the vibrotactile frequency range with a given base amplitude A. The signal is then fed to a simple envelope generator in order to modulate its amplitude. The signal rises exponentially during an attack time t_a, from nil to the original amplitude

A, followed by an exponential decay of release time t_r, mimicking the creation and attenuation of the short and highly damped impact. Last, the modulated signal excites an elementary resonator. For this, a second-order resonant filter is used, creating a resonance peak around a central frequency w_0. The impact signal is therefore approximated as a resonating burst of white noise, with parameters to control its amplitude (A), duration (t_a, t_r) and central frequency (w_0).

4.4.1.2 Control

An impact event is triggered when the distance between a rigid body particle and a fluid particle is below the smoothing radius. Since only the particles at the surface of the rigid body have to be taken into account to avoid false triggers, a new implicit color field c^r is computed only considering rigid body particles: particles belonging to the lowest level sets of c^r belong to the surface. Richardson [153] observed that, in general, the intensity of the impact sound between a rigid body and a fluid is proportional to v^3, where v is the speed of the body at the moment of impact. Hence, after detecting an impact, we can synthesize an impact signal of amplitude A proportional to v^3. We use the same manually set duration and central frequency parameters for all impacts.

4.4.2 Harmonic Bubbles

Small bubbles are generated by small pockets of air trapped under the water surface. Splashes and underwater cavity fragmentation are two causes for small bubble generation. By approximating all bubbles as spherical bubbles and relying on our SPH simulation enhanced with bubble generation, we can easily synthesize and control this component of the model.

4.4.2.1 Synthesis

Following van den Doel [155] and modeling the bubble as a damped harmonic oscillator, the pressure wave $p(t)$ of an oscillating spherical bubble is given by

$$p(t) = A_0 sin(2\pi t f(t)) e^{-dt} \tag{4.2}$$

A_0 being the initial amplitude, $f(t)$ the resonance frequency and d a damping factor.

Minnaert's formula [156] approximates the resonance frequency f_0 of a spherical bubble in an infinite volume of water by $f_0 = 3/r$, where r is the bubble radius. In order to account for the rising in pitch due to the rising of the bubble towards the surface, Van den Doel [155] introduces a time dependent component in the expression of the resonant frequency: $f(t) = f_0(1 + \xi dt)$, with $\xi = 0.1$ found experimentally. Taking into account viscous, radiative and thermal damping, the damping factor d is set to $d = 0.13/r + 0.0072r^{-3/2}$. As for the initial amplitude A_0, previous work [161] suggests, after empirical observations, that $A_0 = \epsilon r$ with $\epsilon \in [0.01; 0.1]$ as a tunable initial excitation parameter. For a detailed explanation of the different hypotheses and equations, we refer the reader to [158] and [155] .

4.4.2.2 Control

Our bubble vibration synthesis algorithm allows the generation of bubble sounds based on two input parameters: the bubble radius r and the initial excitation parameter ϵ. Using our SPH simulation, we couple the vibration synthesis with bubble creation events and automatically select the aforementioned parameters.

Simulating the fluid and the bubbles at a scale matching the smallest bubble radius that generates a perceivable vibration (12mm for a 250Hz frequency) would require around a million particles for a cubic meter of fluid. Achieving real-time performances with this number of particles is currently highly challenging for common hardware. Since we cannot directly link the particle radius to the resonating bubble radius, we transpose the physically inspired approach of Moss *et al.* [158] to determine the radius and excitation parameters: power laws are used both for r and ϵ, within the ranges allowed by each parameter. When a bubble is created, values for r and ϵ are computed, and sent to the signal synthesis algorithm.

4.4.3 Main Cavity Oscillation

During object penetration in the fluid volume, an air cavity is formed. When the cavity is sealed at its surface, it resonates creating a characteristic low-frequency bubble-like sound. Thus, the main cavity can be seen as a large bubble that produces a characteristic low-frequency bubble-like sound. By modeling the cavity as a single bubble with a large radius, we can rely on our harmonic bubble synthesis and control algorithms for this second component of our vibrotactile model.

4.4.3.1 Synthesis

As for the harmonic bubble component (Section 4.4.2.1), we use Equation 4.2 to synthesize the vibration produced by the oscillation of the main cavity. Since we will be using larger values for r, the resulting vibration will be of a much lower frequency, coherent with what we hear in real life. ϵ is set to 0.1 since no variability is desired.

4.4.3.2 Control

In order to detect the formation and collapsing of the main cavity during object penetration, we track the grouping of individual bubbles within our SPH simulation. Bubbles are spawned and stay alive when a cavity begins its closing and collapsing process, until they fill most of the cavity volume, as illustrated in Figure 4.2 and rendered in Figure 4.3 (right, bubbles in blue). At this point, there are bubbles within the cavity that are surrounded exclusively by other bubbles. These bubbles are detected when their color field c^b is above a threshold. If such a particle is detected, there is a potential cavity collapse.

Starting from the detected particle, we perform a search for neighboring bubbles to find the extent of the cavity. Bubble neighbors are added to the set of cavity bubbles, and the process is repeated on the new neighbors until no new neighbor is added. As the search is executed on the GPU, an iterative implementation is required, with one thread per new bubble neighbor, benefiting from our accelerated neighbor search algorithms. During our experiments, we required less than 5 search cycles to account for all the bubbles inside a cavity.

The total number N_b of cavity bubbles is proportional to the volume of the cavity. Since the cavity is modeled as a single large spherical bubble in the signal synthesis algorithm, its radius r can be deduced from the volume of the cavity. Hence, the number of cavity bubbles is mapped to the radius r of the spherical cavity, with user-defined minimal (r^{min}) and maximal (r^{max}) values: $[N_b^{min}, N_b^{max}] \rightarrow [r^{min}, r^{max}]$.

4.5 Vibrotactile Rendering

The vibrotactile model is implemented in PureData, while the SPH fluid and bubble simulation are implemented on GPU, extending the fluid implementation presented in Chapter 2. The communication between the SPH simulation and the acoustic model is handled through the Open Sound Control (OSC) protocol. Each time a bubble, cavity or impact event is detected in the fluid simulation, an OSC message is sent to the acoustic model with the corresponding parameters for sound synthesis.

We designed three scenarios representing three possible interaction conditions. For the graphic rendering, we used a meshless screen-based technique optimized for high frequency rendering, described in Chapter 2. The scenarios were run on a Core 2 Extreme X7900 processor at 2.8 GHz, with 4 GB of RAM and an Nvidia Quadro FX 3600M GPU with 512 MB of memory.

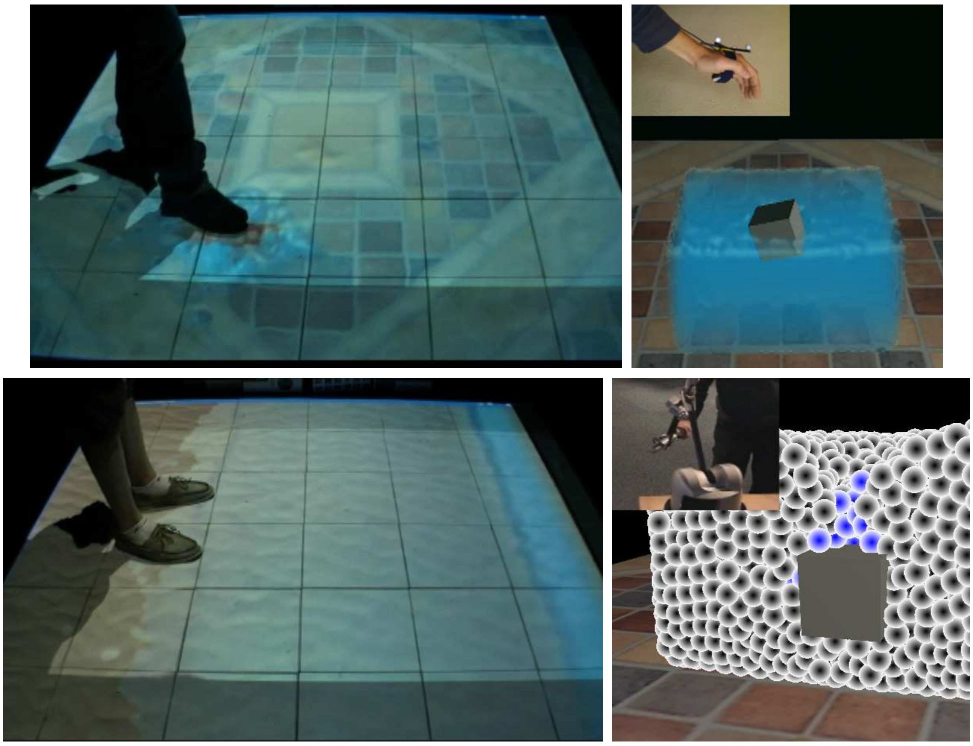


Figure 4.3 – Interaction examples using different body parts and rendering devices: a foot-water pool scenario using actuated tiles (left, top), a user feeling the water under his feet as a wave washes up on the beach using actuated tiles (left, bottom), and a hand-water basin scenario using a small vibrator (right, top) or a 6DoF haptic device (right, bottom).

Active foot-water interaction (shallow pool). Our approach is particularly suited for foot-floor interaction, where the floor renders the vibrotactile feedback to the user's feet through appropriate vibrotactile transducers. We used a floor consisting of a square array of thirty-six 30.5 × 30.5 cm rigid vibrating tiles [31], rendering in the 20-750 Hz range. The VE consisted of a virtual pool with a water depth of 20 cm filling the floor. The user's feet were modeled as parallelepiped rigid bodies and tracked through the floor pressure sensors. The user could walk about, splashing water as he stepped on the pool as seen in Figure 4.3 (left). Performance: 15,000 particles (1% bubbles), 152Hz.

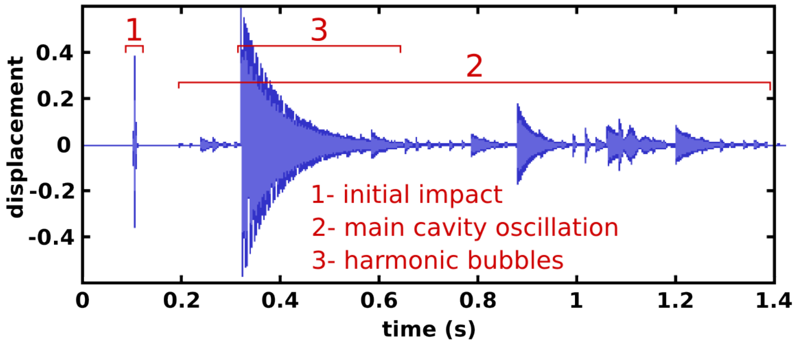


Figure 4.4 – Vibrotactile signal generated with our model during a plunging movement, with its three distinct components: (1) the initial impact, (2) the small bubble harmonics, and (3) the main cavity oscillation.

Passive foot-water interaction (beach shore). Using the same hardware setup as the previous scenario, we designed a tidal action VE in which the user stands still and experiences waves washing up on a sandy beach, as shown in Figure 4.3. Performance: 15,000 particles (6% bubbles), 147Hz.

Active hand-water interaction (water basin). The user can interact with fluids with his hands using a hand-held vibrotactile transducer. In this scenario, a small vibrator was attached to one of the user's hands. The hand was tracked by a motion capture system, and modeled in the VE as a parallelepiped rigid body. He could feel the water sensations by plunging his hand into a cubic volume of fluid, as seen in Figure 4.3 (right). Figure 4.4 shows the vibrotactile signal generated during a plunging movement. Performance: 7,000 particles (6% bubbles), 240Hz.

4.6 Extension to Other Modalities

The model can be combined with existing kinesthetic and auditory feedback techniques to achieve a truly multimodal interaction with fluids.

Kinesthetic Rendering. Kinesthetic feedback can be rendered through a suitable haptic device, such as a multiple degrees-of-freedom force-feedback manipulator. The approach was described in Chapter 2, using the same SPH fluid and rigid body simulation model. This is illustrated in Figure 4.3 (right), which shows a pool of fluid with a rigid body coupled to a Virtuose 6DoF force-feedback device from Haption. The user can interact with the fluid through the rigid body while receiving kinesthetic feedback.

Acoustic Rendering. Since our vibrotactile model is built from sound generation mechanisms, we are able to produce acoustic feedback using the same model, by displaying the signal through a speaker and in the 12 Hz - 20 kHz range. Since the sound synthesis is coupled to a physically based fluid simulator, it enables richer interactions than previous real-time ones, where ad-hoc models (for the filling of a glass) [157] or shallow-water equations [158] were used.

4.7 User Feedback

The novelty of the interaction scenarios, for example, users walking on a volume of fluid and experiencing vibrotactile stimuli, motivated us to design the scenario of a pilot study to assess the perceived interaction qualitatively. The objective of the pilot study was to answer the following questions:

1. Can users recognize that they are stepping onto a simulation of water, despite the contradictory perceptual cues provided by the rigid floor surface?
2. Does the addition of vibrotactile or acoustic rendering improve the realism of the interaction compared to visual feedback alone?
3. How compelling is the overall experience?

4.7.1 Scenario

The scenario required subjects to walk on the virtual shallow pool of fluid described in section 4.5 and perform both a material identification and a subjective evaluation. On the initial presentation, only vibrotactile feedback was provided, with no graphics or audio. Users were asked to identify the simulated material onto which they were stepping. Since the equipment used generates auditory output in conjunction with vibrotactile feedback, we masked the audio by supplying sufficiently loud white noise through four speakers surrounding the floor and headphones worn by the users. The users were then presented three different feedback conditions: visual feedback alone (V), visual + audio feedback (V+A) and visual + vibrotactile feedback (V+Vi), counterbalanced across participants and with three repetitions (nine trials per participant). For each of the three feedback conditions, the users were asked to walk on the actuated floor for 20 seconds and complete a questionnaire, rating each condition on a seven-point Likert scale in terms of *believability* and *engagement* of the interaction.

We gathered the feedback of eight users, all naïve to the purpose of the simulation. An ANOVA on the *believability* question revealed a significant effect of the condition ($F(8, 2) = 3.52$, p-value= 0.045). A post-hoc analysis indicated that the V+Vi condition received significantly higher scores (p-value= 0.042) than the other conditions. An ANOVA on the *engagement* question did not show a significant effect of the condition ($F(8, 2) = 2.48$, p-value= 0.10).

4.7.2 Discussion

The responses to our material identification question, with the presentation of only vibrotactile feedback, were highly encouraging. Six out of the eight subjects stated that they were interacting directly or indirectly with water. This user feedback suggests that our vibrotactile model, based on bubble vibrations, can efficiently convey the sensation of interacting with a fluid volume.

Further insights are available from the details of these responses. Of the six correct answers, two directly identified water, and the remaining four answered that they were walking on “plastic bottles with water inside”, “a floor with wet shoes”, “a water bed”, and “a plank on top of water”. In other words, most of the subjects who associated their experience with water felt that they were interacting with water through a solid material.

We believe the lack of kinesthetic feedback of a sinking foot induced the perception of an intermediate rigid material between the foot and the water. We suggest two possible mechanisms for overcoming this sensory conflict: one involving shoes with vibrating soles to provide the vibrotactile rendering, so that stimuli can be presented prior to contact with the real floor; and a second, employing a vibromechanical actuator, as used in earlier prototypes [16], to deliver the necessary kinesthetic cues of compliance associated with foot-fluid interaction.

As expected, the subjective feedback provided by the users confirmed that the use of vibrotactile feedback greatly improved the interaction experience compared to having only visual rendering. More importantly, the results also suggest that the addition of vibrotactile feedback is more valuable for solid-fluid interaction than audio rendering in terms of the believability of the experience; it also improved engagement, although based on our limited subject pool, the result was not significant. This shows that, despite the lack of kinesthetic feedback to compensate for the conflicting sensory cues provided by the contact of the foot with a rigid tiled floor surface, the effect was perceptually compelling.

4.8 Conclusion

Vibrotactile feedback can provide important additional cues to the user about his surrounding environment when exploring a VE, complementary to visual and kinesthetic modalities. Thus, in this chapter, we introduced the use of vibrotactile feedback as a rendering modality for solid-fluid interaction.

We proposed a novel vibrotactile model based on prior fluid sound rendering knowledge, leveraging the fact that acoustic and vibrotactile phenomena share a common physical source. The model is divided into three components: an initial impact with the fluid surface, a cavity oscillation created when the body enters the fluid, and a set of small bubble harmonics. The model is developed on top of an SPH fluid simulator, enhanced with the ability to simulate bubbles creation and deletion events in a physically based way. Different synthesis and control algorithms are designed to generate a vibrotactile signal for each of the three components of the vibrotactile model.

We illustrated this approach with several fluid interaction scenarios, where users can feel the fluid through vibrotactile transducers using hand or foot-based interaction. Compelling examples included a shallow water pool where users could walk and splash water, a beach shore were users could feel a wave splashing under their feet, and a bucket full of fluid allowing users to plunge their hands inside the fluid volume. Multimodal interaction is also achieved by adapting the vibrotactile model for acoustic feedback, as well as kinesthetic interaction using a previously presented approach.

We conducted an informal study to gather initial user feedback. When subjects were asked to perform material identification solely based on vibrotactile cues, results suggested that the model effectively conveys the sensation of interacting with fluids. The study also highlighted the need for consistent kinesthetic cues during foot-floor contact.

Part II

Infinite Immersive Navigation Based on Natural Walking in Restricted Workspaces

5

Related Work: 3D Walking Interfaces for the Navigation of Large Virtual Environments within Restricted Workspaces

Contents

5.1 Locomotion Interfaces . . . **96**
  5.1.1 Foot-based Devices . . . 96
  5.1.2 Recentering Floors . . . 98
5.2 3D Navigation techniques . . . **102**
  5.2.1 Walking in place . . . 103
  5.2.2 Natural Walking Metaphors . . . 103
  5.2.3 Redirection Techniques . . . 105
5.3 Conclusion . . . **109**

When exploring a VE using immersive VR setups, natural walking is, in many cases, the ideal navigation interface. Several studies have shown the benefits of using natural walking for the exploration of VE, in terms of task performance [20, 21, 22], presence [23] and naturalness [24, 23, 22]. By mapping the real position to the virtual position, natural walking provides the most direct and realistic way of controlling one's position in the VE. In addition, if there is a 1:1 mapping between the real and the virtual motion, vestibular and proprioceptive feedback perfectly matches the visual perception of the virtual movement, producing an ideal multisensory perception of self-motion. Tactile and acoustic cues naturally generated during foot-floor contact also contribute to enhancing the user's immersive experience when using natural walking. Besides, natural walking is, after all, the navigation interface that we use in our everyday life.

However, in most simulations the VE is larger than the real workspace. Immersive spaces such as CAVE-like [121] setups or Head-Mounted Displays (HMD) are inherently limited by the restricted physical space. Users can usually only walk for a few steps before encountering the physical screens or the limits of the tracking space (limited translation). In addition, there is often no rear screen in CAVE-like setups: when the activity in the VE requires the user to turn, the illusion of presence could be broken by the missing screens (limited rotation). Since 1:1 natural walking is not possible, the use of navigation interfaces is therefore required.

There is an important body of work on navigation approaches for VR. Two main categories can be distinguished. The first one, motion interfaces [33], regroups the physical input devices that control the motion of the user in the VE. The second one, 3D navigation techniques, use software solutions to control the virtual motion. They usually involve the tracking of the user's limbs or other body parts, and/or the use of hand-held controllers such as wands.

In this chapter, we first review the motion interfaces, where users exert a walking gait. These are locomotion interfaces [33], some of which enable, to some extent, natural walking within immersive VR setups. The second part of this chapter reviews 3D navigation techniques enabling walking motions of the user.

5.1 Locomotion Interfaces

Hollerbach [33] defines a motion interface as "the mean by which a user travels through a virtual environment". It is the input device that controls the motion of the user in the VE. Motion interfaces can be subdivided in two categories [162]:

- *passive interfaces*. For these interfaces, there is no user locomotive exertion. Typically, passive interfaces are related to a rate control: the user changes his motion rate in the VE by controlling the displacement amount exerted on the device. Common examples are the joystick or wand, where, usually, the greater the angle between the stick and the vertical vector the higher the speed, and a pedal, where speed is proportional to the displacement of the pedal.
- *active interfaces*, or *locomotion interfaces*. With these interfaces, the user is self-propelled through a repetitive gait. Active interfaces directly control the position of the user in the VE. Common examples are treadmills, where the user can walk forward indefinitely while the belt cancels his forward motion.

In the locomotion interfaces category, most of the devices try to enable the use of natural walking. They usually work by compensating the user motion with an inverse motion produced by the device. We do not consider passive interfaces, since we focus on devices enabling natural walking.

Hollerbach [33] classifies locomotion interfaces in four distinct categories: pedaling devices, walking in place, foot platforms and treadmills. However, in the scope of natural walking navigation, we propose a slightly modified classification that we believe to be more appropriate, for two reasons. First, we do not consider walking in place techniques as a locomotion interface, but more as a 3D navigation technique, since they are not related to a specific device, and when devices are used it is only for tracking purposes. Second, many locomotion devices (mainly wearable devices) do not fit into any of the categories mentioned above.

We therefore categorize locomotion interfaces as foot-based devices and recentering floors. Foot-based devices (subdivided in foot-wearables and foot-platforms) are those who compensate the motion of each foot separately, in opposition to the recentering floors that compensate the overall movement. Both classifications are based on how the devices work, not what they allow the user to do.

5.1.1 Foot-based Devices

Foot-based devices are directly linked to the foot. This category can be subdivided in foot-wearables, where the device is mainly a pair of special shoes, and foot platforms, where the device is a pair of moving platforms, acting as a support for the feet, and thus allowing the kinesthetic rendering of additional cues.

5.1.1.1 Foot-wearables

Iwata's Virtual Perambulator [163] is a pair of roller skates and a hoop around the user's waist that limits his movement, as shown in Figure 5.1. The user slides in place while walking, with a body movement closer to a real world movement than passive locomotion. The prototype was improved by using shoes with a low friction film and a brake pad instead of the roller skates. Extensive testing showed that 94% of the tested subjects could freely walk around the VE.

A more recent design from Iwata, The Powered Shoes [164], tries to overcome the need from the user to apply an unnatural extra force when sliding to recenter his position. The recentering is done automatically with roller skates actuated by two motors. The movement of the feet, measured by optical sensors, is compensated by a movement of the active roller skates on the opposite direction. However, the main limitation of this system is that the recentering motion direction is given by the orientation of the shoe, and therefore gaits like side-walking are not possible without awkward foot orientations [165].

Iwata's String Walker [165] is a rather different implementation of a foot-wearable device, since the device includes a turntable surrounding the user to which the shoes are attached through a set of eight strings. The device is illustrated in Figure 5.1. Each string is actuated by a motor-pulley mechanism. The motion of the user is compensated by pulling the shoes in the opposite direction through the strings. The pull is applied to the foot during its stance (foot-floor contact) phase. The device allows omni-directional walking thanks to the turntable that repositions the strings according to the user's orientation. Side-stepping is also possible, following the same mechanism as for forward walking.

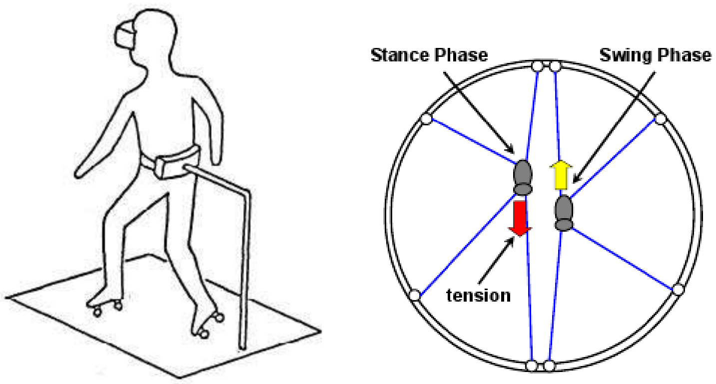


Figure 5.1 – Foot-wearable devices. Left: the Virtual Perambulator [163]. Right: the String Walker [165]

5.1.1.2 Foot platforms

Individually programmed foot platforms allow each foot to be positioned in the 3-dimensional space. The forward motion of the user is countered by a reverse centering motion, keeping the user in the center of the device. The platform can either follow the foot, or apply a force that limits its movement, from a recentering motion to a simulated ground.

The Sarcos Biport (Figure 5.2,left), has two hydraulic 3DoF arms, where the user's feet are attached. The user can also pivot on the foot mounts. The device recenters the user's position translation-wise, but cannot compensate rotations, and the arms can occasionally interfere with each other [166]. It can simulate uneven terrain and smooth surfaces, but allows only moderate walking speeds due to the structural stiffness of the device.

Iwata's Gait Master [166] (Figure 5.2, middle) is made of a pair of 3DoF motion platforms mounted on top of a turntable. The feet, attached to the motion platforms, are recentered by the device, with the user's rotations compensated by the turntable. In the specifications, the Gait Master can be a 6DoF locomotion device. However, a device of only 4DoF was implemented, which is enough to simulate uneven terrains (a set of plain surfaces) but not inclined floors. The device was evaluated on its ability to render an uneven terrain by assessing the naturalness of the user's walk through the measure of the soles pressure on the platforms. The results showed that the force applied at the heel was close to the one on a real stair, and that differences compared to real case values at the toes were due to the use of a safety bar.

The HapticWalker [167] (Figure 5.2, right) is another example of foot platform. It differs from the other devices in its capacity to enable high-speed walking (120 steps per minute) and complex movements (stumbling, sliding), and in its bigger size. It allows a 6DoF motion, but as with the Gait Master a more restricted version was implemented, with 3DoF in the saggital plane, allowing only linear walking. The device was successfully tested by many unexperienced subjects.

Figure 5.2 – Foot platforms. Left: The Sarcos Biport [163]. Middle: The Gait Master [166]. Right: The HapticWalker [167]

5.1.2 Recentering Floors

Recentering floors compensate the overall movement of the user with a recentering motion. This is achieved either actively through motorized floors, which is the case for treadmills and tiles, or passively through user induced motions, such as with spherical environments.

5.1.2.1 Treadmills

A treadmill is a moving platform made of a conveyor belt and an electric motor. A user walking on a treadmill is able to reproduce the gait of walking while standing in place, since the conveyor belt repositions the user at the center of the platform. The main advantage of treadmill-like devices is the realism of vestibular and proprioceptive feedback, since the user is naturally walking forward without constraints. In the literature, we can find 1DoF treadmills allowing only forward and backward motions, and the much more complex omni-directional treadmills allowing rotations and side-stepping as well.

5.1.2.1.a 1DoF treadmills

Linear treadmills, with only 1 linear DoF, are simple but efficient locomotion devices. Figure 5.3 shows the main existing devices in this category.

The Sarcos Treadport (Figure 5.3,left) is a 4 by 8 feet treadmill with a mechanical tether attached to the back of the user's harness. The tether is used for tracking, safety and force rendering purposes. The second generation has a 6 by 10 feet surface and a fast tilting mechanism [168]. The belt speed is proportional to the user's distance to the center of the platform. The ability to apply forces on the user's torso through the tether is used to compensate missing inertial forces, and simulate obstacles in the virtual path. The device allows peak walking velocities of 20 km/h, 1G accelerations, and different body postures, like crawling and crouching [33]. However, turning capabilities are very limited.

The tether can also be used to simulate slopes. Hollerbach *et al.* [169] add a force on the opposite direction of motion when simulating walking uphill, with a magnitude equal to the horizontal component of the force in the real world case. Analogously, the force is applied on the same direction of motion when going downhill. Simulation of side slopes are also possible when applying lateral forces [170].

ATR's ATLAS [171] (Figure 5.3,right) enhances linear treadmills by allowing the user to rotate, and by naturally representing slopes in any direction. The device is a 145 by 55cm linear treadmill mounted on an actuated spherical joint. The joint works like a turntable regarding the user's rotations, and allows the pitching and rolling of the platform in order to simulate slopes. However, rotations are limited, and the treadmill has to be recentered when it reaches its limit in the yaw direction. The user's feet are tracked through markers on the shoes and a video tracking system. The maximum belt speed is 4m/s.

In order to simulate uneven terrain, ATR developed the Ground Surface Simulator (GSS) [172], made of a 150 by 60cm linear treadmill with a deformable belt. Six 25cm long platforms can locally push the belt up to 6cm, as shown in Figure 5.4, at a speed of 6cm/s. A tensioning system ensures the belt fits the changing geometry of the treadmill. The GSS can render a 5°slope.

Taking the opposite approach, Bouguila *et al.* [173] exploit a rotational DoF instead of a linear one. Their device is a 70cm diameter turntable which recenters the user after he has performed a body rotation, so that he is always facing the same direction (always looking at the same display device) in the real environment, hence providing natural vestibular and proprioceptive cues for rotational motions. Although the device recenters the user with a smooth rotation in order to go unnoticed, large and fast rotations compensations might be perceived by the user, generating inappropriate kinesthetic cues. User rotations are tracked through cameras and two optical markers on the user's back. The forward motion is achieved through walking in place. The device was specially conceived with semi-immersive environments (such as CAVEs) in mind, so that users will avoid looking at the missing screens.

5.1.2.1.b Omni-directional treadmills

The main problem of linear treadmills being the impossibility to perform natural rotations, some devices were developed with a 2D motion surface in mind.

The Omni-directional Treadmill [174] tackles the problem by using two orthogonal belts, one inside the other. The outer belt is made of rollers that rotate around the axis perpendicular (in the horizontal plane) to the motion axis of the belt, which is the same axis of rotation as the inner belt. When the inner belt rotates, the rollers rotate with it, therefore creating a motion perpendicular to the motion of the outer belt. The user

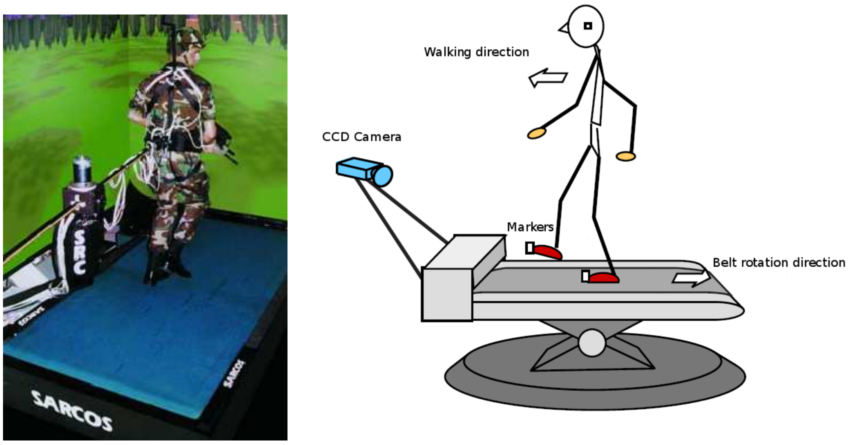


Figure 5.3 – 1DoF treadmills. Left: the Sarcos Treadport (first generation). Right: ATR's ATLAS [171].

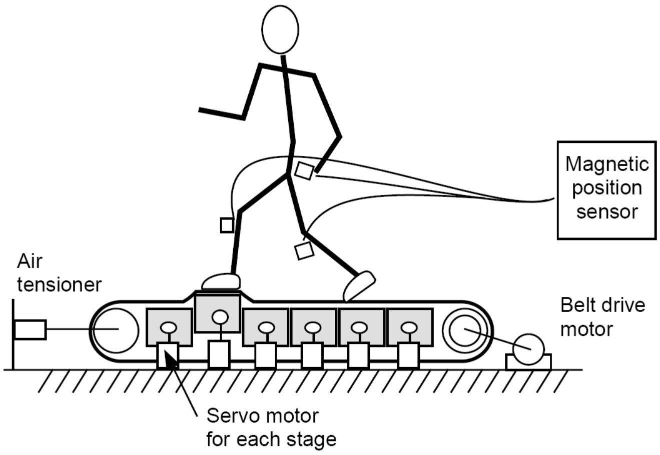


Figure 5.4 – 1DoF treadmills for dynamic terrain: ATR's Ground Surface Simulator [172].

can walk in any direction, since his movement can be compensated in any direction by combining the two perpendicular rotation motions of the treadmill. The usable surface of the device is 1.3 by 1.3m, and the user can walk at up to 2m/s. The main limitations of the device are system lags and inaccurate tracking, which causes the user to stumble. In addition, the walking gait has to be adapted in order to successfully walk on the device.

The Torus treadmill [166] (Figure 5.5, left) is a set of 12 treadmills connected side-by-side, forming a belt that rotates in the opposite direction (in the horizontal plane) to the individual treadmills. The main drawbacks of the device are its usable area (1 by 1m), and the maximum walking speed (1.2m/s) a user can achieve, which forces the user to walk slowly and with short steps.

The Cybercarpet [175] (Figure 5.5, right) uses a different approach. Inspired by the Ball Array Treadmill [176], it has one linear and one rotational DoF, instead of the two linear DoF of the previous devices. A small treadmill, generating the linear DoF, is mounted on a turntable, generating the rotational DoF. An array of balls, set up above the treadmill but independent from the turntable, sums up the two motion vectors, delivering a single motion that recenters the user when walking on top of the ball array. The usable surface has a 80cm diameter, and the user can walk up to 1.5 m/s in translation, and can rotate up to 2rad/s. The considerable friction between the balls and the shoes wears the soles, and the wear debris can degrade the device. Therefore, the user needs to wear shoes with a specific kind of sole material.

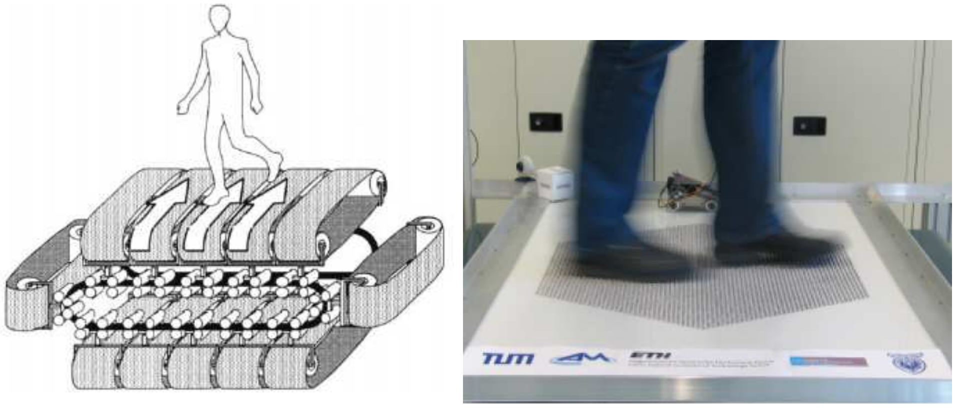


Figure 5.5 – Omni-directional treadmills. Left: the Torus Treadmill [166]. Right: the Cybercarpet [175].

5.1.2.2 Tiles

Recentering floors in the form of tiles provide an alternative to treadmills. The CirculaFloor [177] is a set of tiles with omni-directional motion capability. Together, the tiles form an infinite surface by constantly circulating under the user's feet (Figure 5.6). Through the tracking of the feet, the tile on which the user is standing moves in the opposite direction of the user's motion direction, therefore compensating the motion of the step. Meanwhile, the available free tiles are repositioned on the intended path of the user, in order to compensate his next step. Since the tiles can be positioned around the user in any direction, the user can freely change his motion direction while walking. Although in theory the system allows omni-directional walking, implementations difficulties restrict the user to very slow walking motions.

Figure 5.6 – The CirculaFloor [177]

5.1.2.3 Spherical environments

The particularity of spherical environments, such as the Cybersphere [178] and the Virtusphere [179], is the passivity of the device. It is the user who gives to the floor its recentering motion. As show in Figure 5.7, the user is inside a sphere, and his movement creates a momentum which acts as a recentering motion.

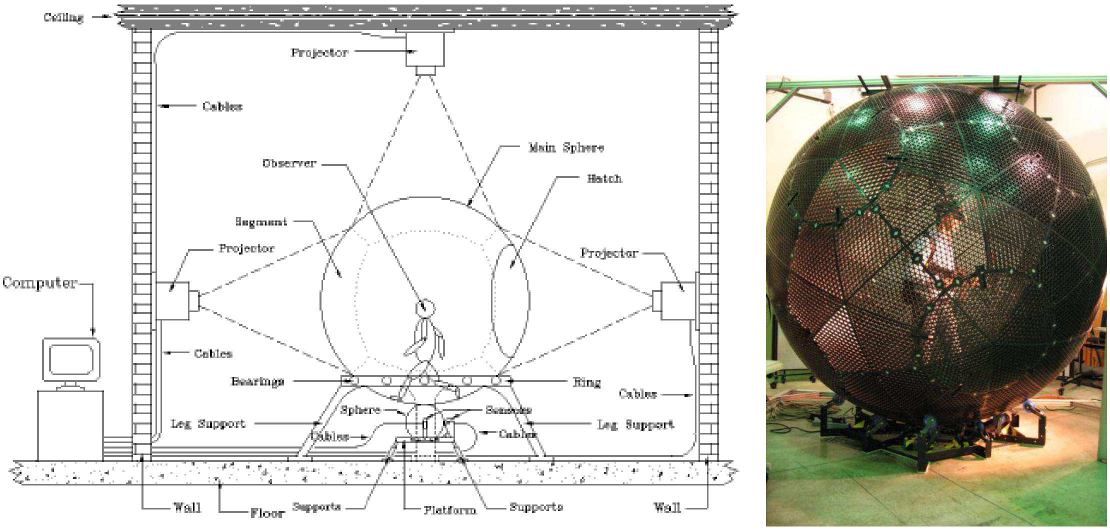


Figure 5.7 – Spherical environments. Left: the Cybersphere [178]. Right: the Virtusphere [179].

5.2 3D Navigation techniques

3D navigation techniques provide a software solution to the problem of navigating a VE, often through a metaphor.

There are several known navigation techniques where the user is not required to walk [12]. Examples of these include *teleportation*, an instantaneous change of position to a new location. In the *Worlds In Miniature* (WIM) [180] metaphor, the user has a copy of the VE in his hands. He can choose a location in the copy and be smoothly taken there in the "real" VE. However, the most common navigation technique is probably the *Flying Vehicle*, where the user can move through the VE using a wand.

As mentioned at the beginning of this chapter, natural walking provides an immersive and intuitive navigation interface for immersive VR setups. Besides, it rids the user from the use of input devices such as locomotion interfaces or wands, and does not require learning. Although static navigation techniques have the advantage of not requiring to deal with limited physical workspace issues, the use of natural walking is preferred. Thus, the main challenge when proposing navigation techniques based on natural walking lies in having an efficient way to deal with physical workspaces smaller than the VE: what to do when the user reaches the boundaries, or how to prevent the user from reaching the boundaries.

We identified three main categories in navigation techniques based on natural walking, that we survey in the remainder of this section. First, walking in place techniques require the user to perform the natural walking gait but without any translation motion (i.e. staying in place). Second, natural walking metaphors combine natural walking with conscious and complementary techniques for dealing with the boundaries. Third, redirection techniques tricks the user into modifying his trajectory in the VE in order to keep him away from the boundaries. Redirection techniques are, to some extent, unnoticeable to users.

5.2.1 Walking in place

The Walking in Place technique [10] allows the user to navigate in a limitless VE without moving in the real environment, but with an increased degree of presence compared to static navigation techniques. By "walking in place", the user simulates the physical act of walking but without forward motion of the body, while a virtual forward motion is introduced. Therefore, the visual optical flow of navigation that should be matched by proprioceptive information from the natural walking gait is matched by proprioceptive information from a gait close to natural walking. The user's sense of presence is greatly increased compared to static navigation techniques, as shown in [23]. However, other sensory data that completes the illusion of walking, mainly vestibular cues, are not perceived by the user.

The Walking in Place technique can be combined to natural walking. The user moves freely in the working space, but whenever he has to go beyond the limits of the physical space he can switch to a Walking in Place navigation. The direction of motion is naturally defined by the user's gaze direction.

Compared to previous techniques where navigation was controlled through the use of the hands (hand-held input devices, gestures), this technique has a major advantage since navigation is entirely devoted to the region of the body that performs that task in the real world. Therefore, what could be done in the real world while walking, can also be done in the virtual one: the hands are free to be used in any other task, such as the manipulation of virtual objects.

In [10], the technique does not require additional hardware, the existing head tracking system is enough. Walking in place is detected through pattern analysis using a neural network, based only on the head position data over time. However, there is a training period of around 5-10 minutes. If walking in place is detected, the user is translated in the VE giving the illusion of real motion. The system yielded a 91% accuracy rate in average.

Additional tracking devices are used in the Gaiter system[181]. Six degrees of freedom sensors are attached to the knees, transmitting translation and rotation values. The Walking in Place function is triggered when the system detects an excess of motion of the legs compared to the normal gait. This can happen when actually walking in place, or when adding extra motion to a real walking motion (hybrid steps). Unlike the previous technique, the direction of motion is defined by the legs motion direction, and not the user's gaze direction, better matching real world locomotion. Hence, the system allows forward, backward, and side-stepping motion.

5.2.2 Natural Walking Metaphors

Natural walking metaphors augment walking with a mechanism to deal with restricted physical workspaces. This mechanism is conscious, meaning that the user is aware that something is being done or asked to be done. However, the goal is to have a seamless transition and an integration of both mechanisms, so that immersion is not broken, and the task can still be performed. Different techniques adopt radically different approaches, as described hereafter.

The Step Wim [34] allows a user to invoke at his feet a miniature version of the VE. Inspired from the World in Miniature navigation (WIM) technique [180], the miniature VE enables the user to travel distances greater than the workspace. Upon invocation, the WIM appears on the ground with the position of the user in the VE corresponding to the position of the user's feet in the real world (Figure 5.8). The user can then walk on the WIM to a new position and trigger a rescaling command that will scale up the WIM until it reaches the VE size. The scaling animation was chosen instead of translation techniques since it improved user's sense of location [34]. The invocation and scaling commands are triggered by bringing together the heels or the toes of the user's shoes. Conductive cloth patches located at the heels and toes of the shoes ensure the detection of the contact between the two shoes.

Figure 5.8 – A user navigating in the VE through the Step WIM technique [34]

Since the Step Wim is implemented in a CAVE setup, the technique solves the limited workspace problem translation wise, but not rotation wise. As in any semi-immersive environment, for a given position and orientation of the user there is a portion of the VE that cannot be displayed. In the case of the CAVE, the main problem are the missing screens. In order to minimize the missing screen problem, the authors of the Step WIM also developed an auto-rotation technique that would ensure that the user would never face a missing wall. The technique is based on a mapping of the 360°of the VE to the system's display field. A linear mapping proved to be uncomfortable since it quickly produced cybersickness. Therefore, the mapping depends on the user's orientation and position inside the CAVE, and is matched by applying a scaling coefficient to the user's head rotation. The closer the user is to the front wall, the larger his field of view. There, the rotation scaling is not as large as if the user was at the edge of the CAVE, where the field of view is the narrowest. The technique is noticed by the user, who needs a small adaptation time in order to adjust to the new mapping, and does not entirely solve the missing screen problem.

Working with HMD and large VE, Williams *et al.* [35] explored different techniques to overcome the limited workspace problem when using natural walking for navigation. They proposed and compared 3 different techniques that would reset the user's position in the real world when he reaches the workspace limits, without breaking his spatial awareness of the VE. The main idea is to warn the user when he is about to reach the limits, so he can execute the resetting motion and keep traveling in the same direction he has been traveling in the VE. In the *freeze-backup* technique, the VE is frozen and the user takes steps backwards to recenter his real world position inside the workspace. In the *freeze-turn* technique, the orientation of the user is frozen while he physically pivots 180°, while in the *2:1-turn* technique a 360°virtual rotation is mapped to a 180°real world rotation, and the user also physically pivots 180°. For each technique, the spatial awareness was evaluated according to two criteria: the angle error when asked to face, eyes closed, a particular object after a reset, and the time the user took to face the object. Results showed that the freeze-backup technique had the lowest angle error, making it better suited for applications where spatial orientation is important, while the 2:1-turn had the lowest time, more appropriate for applications where time is a constraint. Both were superior to the freeze-turn technique.

The Seven League Boots [182] is another technique to naturally walk in VE larger than the workspace. It relies on the scaling of the user's speed only along his intended direction of travel. In order to detect the intended direction of travel, an approach considering both the user's gaze direction and the direction of previous displacement over the last reasonable period of time is proposed. Both criteria are weighted according to the amount of displacement over the last period of time: if the user stood still or did not travel much, the gaze criteria will be weighted higher, but if the user showed a will to move, the displacement will be weighted higher. However, the experiments conducted in order to evaluate the Seven League Boots used a simpler version of the technique, since the intended direction of travel was pointed by the user through a wand. The goal of the experiments was to show how comfortable it was for the users to have their speed scaled only on the intended direction of travel. Therefore, the simpler version was compared to natural walking, natural walking with uniform gain, and wand flying. Results showed that, overall, the Seven League Boots were more appreciated than the other techniques. No quantitative evaluation measuring spacial awareness was reported.

5.2.3 Redirection Techniques

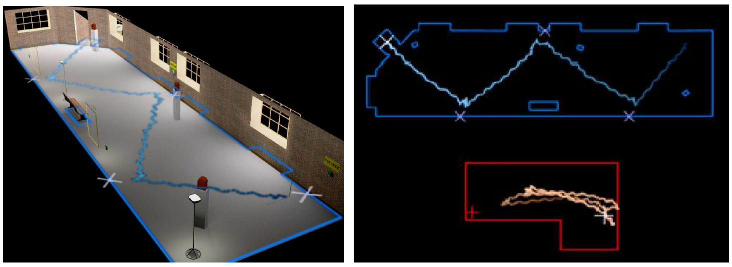
Redirection techniques introduce changes to the user's point of view or to the VE in order to trick the user into changing his trajectory and thus avoid reaching the boundaries of the physical workspace. Different approaches exist, depending on what is manipulated and how it is manipulated.

5.2.3.1 Redirected Walking

When experiencing Redirected Walking [36], a user is tricked into walking in a curved path in the real world when walking in a straight line in the VE. In a sufficiently large workspace, and with a straight virtual path, the user can naturally walk endlessly without reaching the limits of the real workspace, and without noticing that he is actually walking in circles. The technique relies on the interactive and imperceptible rotation of the VE around the user, so that he is always walking towards the farthest side of the real workspace. The "redirection" takes into account the user's position, orientation and speed. However, in small workspaces such as CAVE setups it is not possible to keep the user away from the walls by injecting only unnoticeable amounts of rotation distortion: there is a trade-off between workspace size and user perception of the rotations. Such a trade-off can be avoided by using controlled events that force the user to look around him, hence allowing the injection of additional rotation distortion.

In a pilot user study, Razzaque *et al.* [36] showed that the technique worked well in a 4 by 10 meters workspace with a virtual room twice the size and a predefined path. Figure 5.9 shows a zig-zag path in the VE. Users had to follow that virtual path and press a button at each turning point. They were told to look at the next button once they had pushed the one next to them. The systematic rotation of the user to face the new button was the occasion to inject the rotation distortion. Then, if needed, more rotations could be injected during the walking phases. Subjects could complete the test without noticing the extra amount of rotation or running into walls. However, the path was predefined: the user could not freely wander about the room.

Since Redirected Walking relies on the unnoticeability of the redirection, Steinicke *et al.* [183] statistically quantified the maximum amount of rotation distortion that could be injected without being noticed by the user. In addition, they allowed the scaling of translational movements, and quantified the maximum scaling that could be introduced without altering the user's perception of his virtual self motion. After a user study testing several users



(a) (b)

Figure 5.9 – The VE used to test the Redirected Walking technique [36] (a) with the corresponding user paths, virtual (blue) and real (orange) (b)

with different rotation and translation gains, results showed that, statistically, rotations scaled down or up to 30% and translations scaled down to 15% and up to 45% are perceived in less than 20% of all cases. Therefore, a user walking on a straight virtual path can be redirected into a 3.3 meter radius circle. However, these results are subjective, varying from a user to another.

Peck *et al.* [184] introduced the use of distractors in order to force the user to rotate his head and allow the injection of rotation distortion. They compared Redirected Walking coupled to a distractor technique (a flying moving sphere) with other reorientation techniques such as direct VE rotation, mapping of the 360°of the VE to 180°in the real world and Redirected Walking with audio instructiosn to force the rotation of the head. The VE was a 200 meter straight path with checkpoints every 5 meters, while the real workspace was a 5 meter straight path. At every checkpoint, the users were presented one of the four reorientation techniques, which made them rotate around themselves in the real world, but kept them in the same direction of motion in the VE. Results showed that the distractor and the instructed head rotation techniques were rated significantly higher than the others, in terms of presence, preference and naturalness. In a second experiment with different subjects, a butterfly distractor was compared to the sphere distractor and the instructed head rotation, showing a preference for the more “natural” distractor technique.

Redirected Walking has been used in applications beyond its original goal of natural walking navigation. Razzaque *et al.* [185] explored the use of Redirected Walking with a walking in place technique in a CAVE setup. The goal was to avoid looking at the missing screen of the CAVE when exploring a VE. By comparing the technique with a hand-held navigation technique, they showed that the frequency of looking at the missing screen was not reduced, although the variance was. In a different context, Kohli *et al.* [186] showed how to take advantage of the Redirected Walking technique to use passive haptics (simulating the haptic feedback of a virtual object through the use of a real object) on multiple identical virtual objects with only one real object. Figure 5.10 shows the tested VE superimposed to the real world. A distractor (a moving target that had to be neutralized) was triggered at specific synchronization zones in order to make the real object coincide with the virtual object faced by the user. As with the previously described Redirected Walking applications, the path in the VE has to be predefined and carefully chosen.

In the specific case of architectural virtual walkthroughs, the *Arch-Explore* interface [187] allows the use of redirection techniques in small workspaces such as CAVEs. A virtual door mechanism is combined to increased (perceptible) redirection gains and the splitting of virtual rooms considered too large for redirection into smaller sized rooms. The missing screen problem is not addressed in this work.

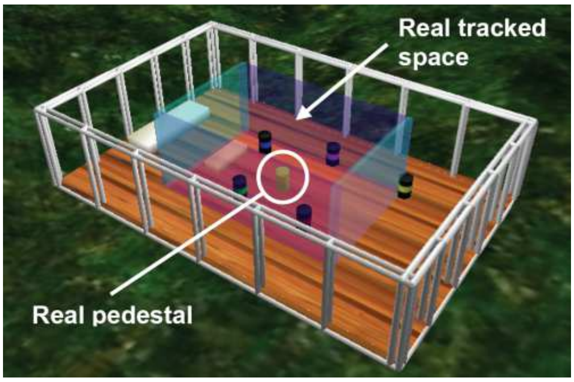


Figure 5.10 – The VE used to test passive haptics with the Redirected Walking technique [186]

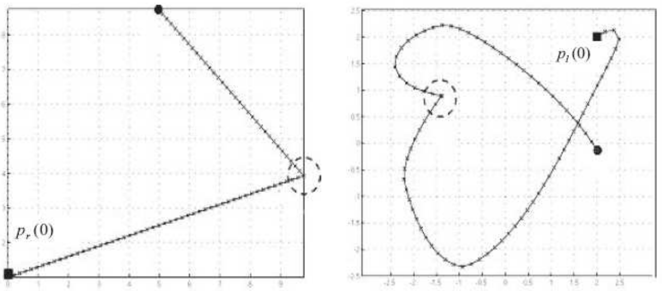
5.2.3.2 Motion Compression

Along the same idea as Redirected Walking, Motion Compression [188] rotates the VE, keeping the user away from the limits of the workspace and allowing him to explore a VE larger than his workspace. However, Motion Compression differs from Redirected Walking in many aspects. First, rotations and translations are mapped with a 1:1 ratio: no rotation (or translation) is injected, only the VE is rotated. Hence, distractors cannot be used. Second, the unnoticeability of the rotations is not a constraint: the user can perceive the manipulation, specially when close to the limits, where it is simply not possible to bend the user's path by unnoticeably rotating the VE. And third, the user's path in the VE does not need to be predefined: without the unnoticeability constraint, paths can be greatly curved, giving to the user a higher freedom of movement. Instead, the path is predicted in real time. Figure 5.11 shows an example of Motion Compression.

There are three steps in the Motion Compression algorithm: path prediction (finding the intended virtual path of the user), path transformation (mapping the intended virtual path to a real path inside the boundaries of the workspace, with minimum curvature and preserving distances and angles) and user guidance (applying the rotations as the user walks the path). In Nitzsche *et al.* implementation, path prediction is based on the user's gaze direction and a set of manually selected potential targets. A potential target is activated as the actual target if it stayed the longest inside the user's field of view among the other potential targets. The more a user looks at a potential target, the higher the probability the target is the actual path target. Path prediction is performed at every update of the tracking system. However, it requires the availability of a model of the VE. Path transformation is done by minimizing the total curvature deviation.

In a first experiment, Nitzsche *et al.* tested different curvature deviations in order to find the perceptual threshold under which subjects could not tell the direction of the curvature. The threshold was found equal to $0.1m^{-1}$. In a second experiment they simulated a 30 by 20 meter virtual museum hall, with a 3 by 3 meter real workspace. With a curvature deviation usually greater than $0.3m^{-1}$ due to the small workspace, the virtual visit was a positive experience for most of the users regarding comfort and intuitiveness.

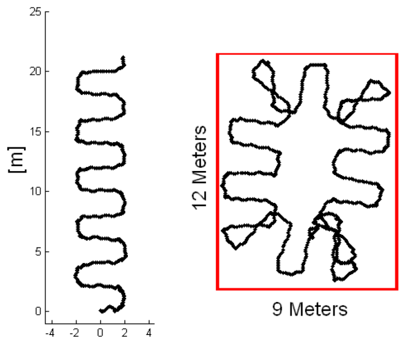
Engel *et al.* [190] proposed a different technique for path transformation by including a perceptual component. Instead of minimizing the curvature deviation, they look for a path that both maximizes the distance to the boundaries of the workspace and minimizes the discomfort of the user. A perceptual study allowed the construction of a psychometric function that gives, for each rotation gain, the probability of noticing the gain. The function is then translated into a cost function representing the level of discomfort of the



(a) **(b)**

Figure 5.11 – (a) The virtual path followed by a user and (b) its real counterpart computed through Motion Compression [189]

user for each rotational gain. Figure 5.12 shows an example of a virtual and a real path followed by a naïve user.

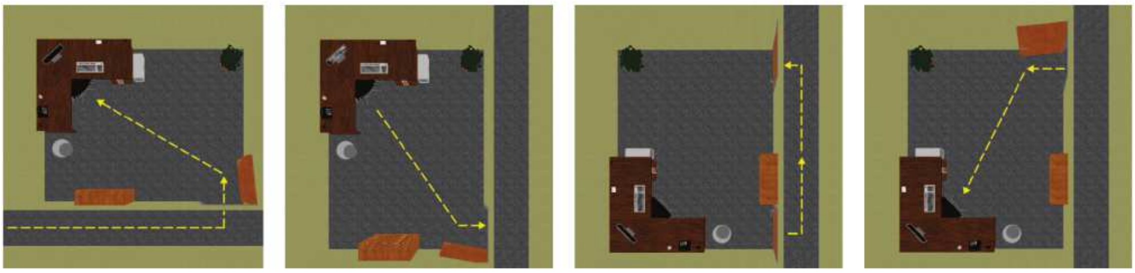


(a) **(b)**

Figure 5.12 – Another example of Motion Compression: (a) the virtual path (b) the real path [190]

5.2.3.3 Change Blindness

In recent work, change blindness redirection [191] redirects users wearing HMDs by making dynamic changes to the VE. Examples of VE manipulation are the instantaneous rotation of a doorway, and the realignment of the corridor behind, as shown in Figure 5.13. The manipulation is performed when the user is focusing on a distracting task. The main advantage of this technique is that the manipulation is not noticed by the users. Hence, users can navigate in very large VE (220 m^2 in [191]) within relatively small tracking areas (4.3 feet in [191]), provided the VE was designed for the technique as a series of small rooms. The technique is further developed in [192] by introducing passive haptic feedback, like Kohli *et al.* [186] did for Redirected Walking. The real workspace has a floor with one gravel zone and one concrete zone, while the VE has several gravel and concrete zones in different rooms and outdoor paths. Using change blindness, the real floor zones can be reused in the different places of the VE, without the user knowing that he is stepping on the same ground over and over.



(a) The user enters a 14' x 11' room and walks towards a desk to view the computer monitor.

(b) When the user approaches the monitor, the corridor and the door to exit the room are instantaneously rotated by 90 degrees.

(c) When the user enters the hallway, the second doorway is added, and the contents of the room are swapped with the next room.

(d) The first door is removed as the user enters the second room. The process can then be repeated.

Figure 5.13 – Sequence of changes exploiting change blindness, allowing the user to navigate in a VE large than the real workspace without noticing it [191]

5.3 Conclusion

In the perspective of studying navigation techniques from the exploration of VE larger than the walking workspace, this chapter presented a synthetic survey of the existing hardware (locomotion devices) and software (3D navigation techniques) solutions. We first proposed a classification of existing locomotion devices into foot-based devices and recentering floors. We showed that most of the devices try to enable the use of natural walking by compensating the user motion with an inverse motion produced by the device or through the device. Although a very promising solution, we can notice that most devices exhibit limitations, in terms of walking and turning speed, accuracy and user stability, and thus fail at fully restoring kinesthetic and proprioceptive walking cues. In addition, most of them are bulky, heavy and expensive, and have yet to be adopted outside the research labs.

We then focused on 3D navigation techniques, surveying walk in place techniques, natural walking metaphors and redirection techniques. All three categories use radically different approaches to allow the navigation beyond the physical boundaries: using a natural walking gait without forward motion, using a metaphor that takes into account the workspace boundaries, and tricking the user into walking on specific paths that avoid reaching the boundaries without him noticing. However, through this overview of 3D navigation techniques, and considering the objectives of this Ph.D. thesis, several issues stand out.

First, there is clearly a lack of adequate navigation techniques based on natural walking allowing the infinite exploration of small workspaces, such as CAVE-like setups or HMD setups with limited tracking range. Walk in place techniques avoid reaching the boundaries only if the user always remains in place, which is a rather unnatural constraint that limits interaction possibilities and immersion. Most walking metaphors do not avoid reaching the boundaries and do not provide safety mechanisms, while those that do avoid the boundaries generate breaks of immersion. Redirection techniques are very promising, but only work well in large workspaces, often with defined paths or specific real and virtual environments. We address this issue in Chapter 6, when we present an immersive walking metaphor for infinite navigation in restricted workspaces.

Second, the rotation boundary issues are not addressed in previous work. There is no system designed to avoid breaks of immersion due to users looking in the direction of the missing screen. The only exception is a walk in place technique combined with redirection, but results showed that users did not look less at the missing walls. Thus, this remains an open issue: there is no natural walking navigation technique fully adapted to CAVE-like setups. We address this in Chapter 7, where we propose immersive walking metaphors taking into account translation and rotation boundaries.

6

Infinite Navigation in Large Virtual Environments within Restricted Translation Workspaces

Contents

6.1 The Magic Barrier Tape **112**
6.1.1 Display of the Workspace Limits 113
6.1.2 Navigation Through Rate Control 113
6.2 Extending Resetting Techniques for Omni-Directional Walking **115**
6.2.1 Extended Freeze-Backup Technique. 115
6.2.2 Extended 2:1-Turn Technique. 116
6.3 Evaluation **116**
6.3.1 Experiment #1: Pointing Task 116
6.3.2 Experiment #2: Path Following Task 119
6.3.3 Subjective Questionnaire 122
6.4 General Discussion **122**
6.5 Conclusion **124**

When navigating in a VE, natural walking provides the most natural, intuitive and direct way of controlling one's position without removing vestibular and haptic cues. Several studies have shown the benefits of using natural walking for the navigation of VE, in terms of task performance [20, 21, 22], presence [23] and naturalness [24, 23, 22].

However, in most simulations the VE is larger than the real workspace. Navigation techniques become, paradoxically, both a strength and a weakness of current VR systems. They allow individuals to easily and instantaneously travel long distances and to follow impossible virtual paths. However, they are unable to fully reproduce real life infinite walking capabilities since the user can quickly reach the boundaries of his real workspace, leading to safety and immersion issues. Hence, the illusion of a VE is lost through the use of unnatural navigation techniques usually coming from the 2D realm.

In this chapter, we propose a natural metaphor for navigation in restricted size workspaces, using natural walking in position control when inside the workspace, and an interaction technique in rate control at the limits of the workspace. The main idea is to use a well-known real world object, the barrier tape, and its well-known association to the "do not cross" message. The technique visually and clearly defines the workspace where the user can freely walk by surrounding it with virtual barrier tape. When the user reaches the virtual barrier tape, he can move farther in the VE by “pushing” on the virtual barrier tape. Hence, the technique allows the navigation in an unlimited VE, allowing natural walking

Figure 6.1 – The Magic Barrier Tape displays the boundaries of the real workspace as a virtual barrier tape, and uses a hybrid position/rate control to travel in the VE. The user (left) "pushes" on the Magic Barrier Tape (center) to move inside the VE when he reaches the workspace boundaries. Any tracked body part can be used to trigger the Magic Barrier Tape (right).

when inside the workspace boundaries, providing an environment safe from collisions with the displays or tracking data loss, and this in a natural and efficient way, without break of immersion. The technique is evaluated by comparing it with other techniques providing safe navigation.

This chapter is organized as follows. The design and implementation of the Magic Barrier Tape are presented in Section 6.1. We then describe extensions made to existing techniques fulfilling the same goals in Section 6.2, in order to compare our contribution with the state of the art. We then report on the experiments conducted for the evaluation of the Magic Barrier Tape in Section 6.3, and present a general discussion on the results of the experiments in Section 6.4, before concluding.

6.1 The Magic Barrier Tape

We propose a novel interaction metaphor, the Magic Barrier Tape (Figure 6.1), that provides a solution to immersive infinite walking in a restricted size workspace through a natural and efficient metaphor.

Walking workspaces of VR systems are often bounded by the tracking area, the display devices or by the walls of the immersive room. Hence, the Magic Barrier Tape has two fundamental objectives. The first one is to inform and display the limits of the workspace in a natural way, without break of immersion, in order to avoid the collision with physical objects outside the workspace boundaries or leaving the tracking area. The second one is to provide an integrated navigation technique to reach any location in the VE, beyond the walking workspace.

To overcome the mismatch between the restricted size workspace and the potentially infinite size of the VE, we followed the concept of hybrid position/rate control [193], used in a different context for object manipulation, where position control is used inside the available workspace for fine positioning, while rate control is used at the boundaries for coarse positioning. This concept can be found in common desktop applications and games where the mouse switches to rate control when it reaches the edge of the screen: in a file manager when doing multiple selection, or in top-view strategy games such as Starcraft when panning on the map. In our context, we applied the concept to navigation, with the available workspace being the walking workspace. The boundaries of the workspace are represented by a virtual barrier at mid body height textured with slanted black and yellow stripes, evoking the use of barrier tape and its implicit message: "do not cross".

The real workspace, delimited by the physical boundaries, is mapped to a virtual workspace inside the VE, delimited by the virtual barrier tape. Inside the workspace, we use position control: the user can freely walk, and objects inside the virtual workspace can be reached and manipulated through natural walking and real life movements. When reaching the boundaries of the workspace, we switch to rate control: the user can move farther in the VE by "pushing" on the virtual barrier tape, hence translating the virtual workspace in the VE. He can then perform a task inside the virtual workspace at the new location.

The Magic Barrier Tape concept is not subject to a specific technology. It can be implemented in many different VR systems. Any object or body part can be used as an actuator for the virtual barrier tape, depending on the application, and the rate control law can be fitted to specific behavioral needs. In the remaining of this section, we detail the Magic Barrier Tape concept. We take as implementation example our own VR setup, consisting of a Head Mounted Display (HMD) with a 1.5m radius cylindrical tracking space, and one of the user's hands as actuating object.

6.1.1 Display of the Workspace Limits

The boundaries of the workspace are displayed through 3 complementary visual cues: the main virtual barrier tape, the warning virtual barrier tape, and their gray shadow on the floor, as shown in Figure 6.2.

The main virtual barrier tape is presented as a band that matches the shape of the workspace boundaries, such as a square for a CAVE or a circle for a cylindrical tracking system. It is positioned at a safe distance ahead of them, high enough from the virtual floor so that the user does not need to look down to see the barrier tape, and low enough so that it does not occlude the user's forward vision. The boundaries of the workspace are therefore clearly and continuously visible. The tape is made slightly translucent so what would have been normally hidden by the tape is still discernible.

The warning virtual barrier tape appears when the user's body is close to the main tape, as a warning signal. This second tape has the same shape and origin as the main one, and has a red glow to capture the user's attention. For the same reason, it is positioned at the user's eyes height. The tape is fully transparent when the user is at a reasonably safe distance from the main tape, and becomes progressively opaque as the user gets closer, therefore making the warning signal also progressive, from dim to strong. The warning virtual barrier tape is complementary to the main tape, since it is triggered as a safety measure, and it gives an idea of when to stop walking and start "pushing".

The tapes shadow is drawn on the floor as if the barrier tapes were lit from above, in order to have a visual cue about the limits of the workspace when the user looks down. Hence, at least one of the 3 visual components is always visible at almost any viewing direction, which is particularly helpful with an HMD setup where there is usually a narrow field of view.

In our VR setup implementation, the main virtual barrier tape is 30 cm high and at 30 cm from the boundaries. It is shaped as a ring with a 1.2m radius and the center of the tracking area as origin. It is positioned at 1.3m from the virtual floor. The warning tape is activated when the user is at 30 cm from the main tape.

6.1.2 Navigation Through Rate Control

The Magic Barrier Tape allows the use of position control inside the workspace, and rate control at the boundaries. The user is switched from position control to rate control

Figure 6.2 – The three Magic Barrier Tape visual cues to show the workspace boundaries: the main virtual barrier tape (middle), the warning tape (top) and the tapes shadow (bottom).

whenever his hand (or any other tracked body part) penetrates the boundaries represented by the virtual barrier tape. The speed of the resulting translation in the VE is a function of the hand penetration distance. When the user's hand is pulled back inside the workspace, the user is switched back to position control.

The virtual barrier tapes (main and warning) are deformed when the user's body (preferentially, the hand) penetrates the boundaries. This elastic behavior allows the user to see how deep he is "pushing", and therefore to evaluate how fast he will move in the VE. A visual feedback on the rate control is also important so the user can know where the neutral position is located [193].

The deformation follows the shape of a centered Gaussian curve D, of equation:

$$D(p) = p\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}$$

where p is the penetration length (in meters), and σ the standard deviation, which controls the "width" of the deformation. The virtual barrier tape is rotated so that the center of the Gaussian curve matches the penetration point P, the collision point between the hand and the virtual barrier tape. Therefore, as shown in Figure 6.3, the Gaussian deformation is centered around the penetration point, and its symmetry axis is given by the $\overrightarrow{OP}$ direction, where O is the center of the virtual barrier tape. Since the deformation follows the user's hand, the Gaussian curve has to be shifted to take into account the lateral deviation of the hand position H with respect to the Gaussian axis, as shown in Figure 6.3. The final result gives the impression of having an elastic region around the penetration point than can be deformed in any direction. This deformation direction, $\overrightarrow{PH}$, gives the travel direction of the virtual workspace (Figure 6.3).

The velocity V, a function of p, gives the speed of travel. It has the following equation:

$$V(p) = k * p^n$$

where k and n are constants. We use a polynomial function in order to have both slow speed when the user is close to the boundaries for small distances, and high speeds to move fast for distant targets. In our implementation, after preliminary testing, we used $\sigma = 0.15$, $k = 1.4$ and $n = 3$.

Our Magic Barrier Tape implementation provides both a safe walking environment and a natural and efficient navigation technique.

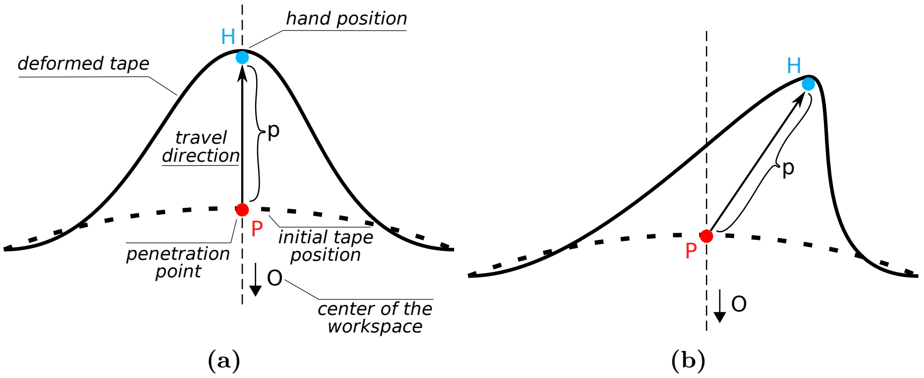


Figure 6.3 – The Gaussian deformation of the Magic Barrier Tape in top-view (a) and its shifted version (b) to follow the hand position.

6.2 Extending Resetting Techniques for Omni-Directional Walking

In order to conduct the evaluation of the Magic Barrier Tape, we chose similar purpose techniques among existing navigation techniques. Among the surveyed active navigation techniques, based on natural walking, only the resetting techniques developed by Williams *et al.* [35] (see Chapter 5), namely the Freeze-backup and the 2:1-Turn techniques, provide collision free and infinite navigation capabilities. However, the resetting techniques were originally designed for straight paths and right angle turns, whereas in most VR applications the user is allowed to freely explore his surrounding VE, taking arbitrary paths and freely rotating around him. For fair comparison throughout the evaluation, since our Magic Barrier Tape technique enables such a navigation, we propose to add visual cues to these techniques in order to make them well suited for omni-directional navigation.

6.2.1 Extended Freeze-Backup Technique.

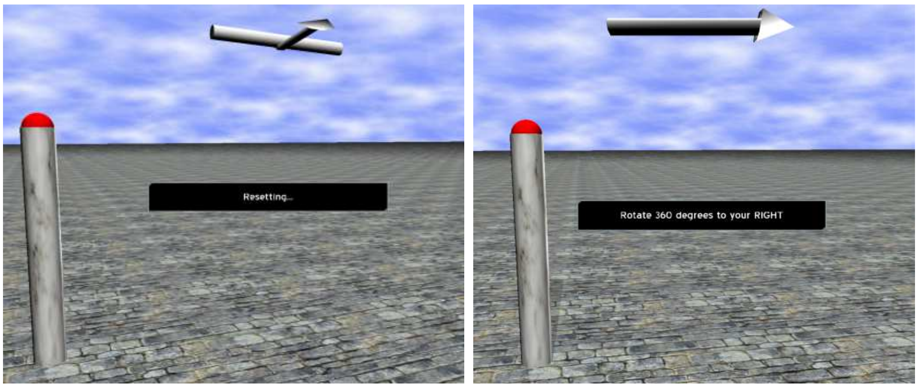
In the original Freeze-backup technique, in order to reset his position the user has to walk backwards in a straight line, until he reaches the resetting position. Since he is not guided while walking backwards, paths can only be straight. Otherwise, he could reach the workspace boundaries prematurely and find himself "locked" in a very short path resetting loop.

In the extended Freeze-backup technique, backups now need to take the user to the center of the real workspace. Before the reset, the user can be at any position in the real workspace, and with any orientation. Hence, we propose to add visual cues to guide the user through his resetting motion, which is divided in two steps. First, the body needs to be oriented towards the resetting position. An horizontal segment is drawn on the screen representing the user's orientation with respect to the resetting position, like his shoulder line seen from above in the real workspace reference frame. The user has to change his orientation until the segment becomes parallel to his body. Then, as a second step, the user has to walk to the resetting position by following an arrow direction. The arrow becomes smaller as the user gets closer to the resetting position, indicating how far he is from his target. Through this mechanism, the user can reach the center of the real workspace from anywhere in the real workspace. Figure 6.4a shows the segment and the arrow drawn at the top of the screen.

6.2.2 Extended 2:1-Turn Technique.

In the original 2:1-Turn technique, a 180° real rotation of the user is mapped to a 360° virtual turn, and the user stays on the same real path but on the opposite direction. Since real turns are always of 180°, walking paths need to be straight with eventually right angle turns to avoid the same "locking" problems mentioned above.

In the extended 2:1-Turn technique, real turns can no longer be of only 180°. The resetting angle is given by the non oriented angle between the viewing direction and the body position - resetting position vector. The virtual angle remains the same, 360°. For any orientation before resetting, two turning directions are possible: to the left and to the right. To each direction corresponds an angle, with usually one greater than the other. The direction with the largest angle is chosen, so that the rotation gain when mapping the turn to a 360° virtual turn is lower, and the illusion is therefore less perceivable. As show in Figure 6.4b, an arrow drawn at the top of the screen indicates the turning direction to the user.



Resetting...

Rotate 360 degrees to your RIGHT

(a) (b)

Figure 6.4 – The visual cues (top of the screen) from: (a) the extended Freeze-backup technique, giving the shoulder orientation and the walking direction, and (b) the extended 2:1-Turn technique, giving the rotation direction. The black panels read "Resetting..." for the extended Freeze-Backup, and "Rotate 360 degrees to your RIGHT" for the extended 2:1-Turn.

6.3 Evaluation

In order to demonstrate its suitability for infinite navigation within a restricted workspace, we evaluated the Magic Barrier Tape by comparing it to two other existing navigation techniques that enable collision free infinite walking within a restricted workspace, namely the Freeze-Backup and the 2:1-Turn resetting techniques [35] with our extensions for omnidirectional walking. We conducted two experiments, a pointing task and a path following task.

6.3.1 Experiment #1: Pointing Task

In Experiment #1, our goal was to compare the 3 techniques over a pointing task where the user had to move from a central initial location to a new location, indicated by a

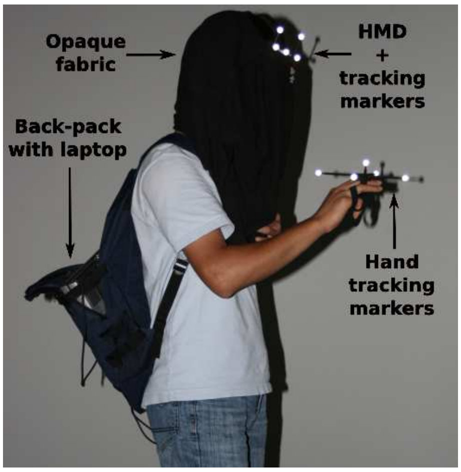


Figure 6.5 – A subject wearing the tracking equipment, the opaque fabric for occlusion and the backpack with the laptop computer.

target, as fast as possible. We a priori assumed that the Magic Barrier Tape will be faster, since rate control allows speeds greater than the average walking speed.

6.3.1.1 Description

6.3.1.1.a Population.

Twelve participants (1 female and 11 males) aged from 24 to 59 (mean = 30.3, sd = 5.7), took part in this experiment. Two of them were left-handed, and none of them had known perception disorders. They were all naïve to the purpose of the experiment.

6.3.1.1.b Experimental Apparatus.

The experiment was conducted in a closed room with dim light. We used the eMagin Z800 Head Mounted Display as display device, at 60 Hz and with stereoscopy enabled. The user was wearing an opaque fabric on top of the HMD to avoid seeing the surrounding real world. The user was carrying a backpack with the laptop computer running the application, and could therefore move freely (Figure 6.5). The user's head and hand were tracked by an ART ARTtrack2 infrared tracking system with 9 surrounding cameras for 360° tracking. The available tracking space was a cylinder with a 3m diameter and a 2.5m height.

The VE consisted of a flat infinite floor with a rock floor texture, a cloudy blue sky, and a target made of a 1.4m high and 0.2m radius marble textured cylinder with a 0.2m radius red hemisphere on top, as illustrated in Figure 6.6.

6.3.1.1.c Procedure.

Before the beginning of the experiment, the three techniques were explained to the subject. Before each trial, the subject had to go back to the initial position and orientation at the center of the workspace. Then, the experimenter launched the next trial. The participant was instructed to look for the target in the VE and to walk towards it until collision. For each technique, he had 3 trials for training. The experiment lasted about 30 minutes, including training trials.

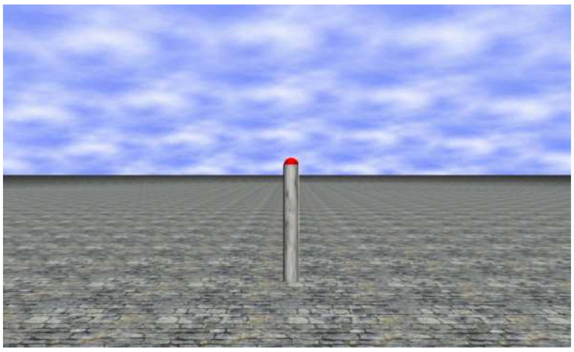


Figure 6.6 – VE used in Experiment #1 with a target.

6.3.1.1.d Experimental Plan.

Participants completed all the three technique conditions (Barrier Tape, Freeze-backup, and 2:1-Turn) and the order of the conditions was counterbalanced across participants. In each condition, the participants were exposed to 3 successive blocks of 6 trials (2 different distances × 3 different directions). The 3 possible directions are at 120° each, one of them being the user's initial direction, and the two possible distances are 2.2m and 5m. In each block, the presentation order of these trials was randomized. Participants completed a total of 54 trials (6 target positions × 3 technique conditions × 3 trials per condition). During a learning phase, prior to each technique condition, participants were exposed to 3 trials that did not enter in the final data set.

6.3.1.1.e Collected Data.

For each trial and each subject, we recorded the completion time (in seconds) and the amplitude of walking in the real world (in meters). The completion time is the time took by the subject to complete the trial. The amplitude of walking in the real world corresponds to the distance traveled when walking in the real *and* the virtual environment. It is the movement in the real world that makes the user move forward in the VE (as opposed to a resetting movement where the user moves in the real world, but not in the virtual one).

6.3.1.2 Results

For the different comparison analyses, a correction for experiment-wise error was realized by using Bonferroni-adjusted alpha level ($p = 0.05$ divided by the number of tests). Thus, in order to compare the Barrier Tape technique to the two other techniques (Freeze-backup and 2:1-Turn) the alpha level was adjusted to $p = 0.025$.

6.3.1.2.a Completion Time.

Using the completion time data collected during the experiment, we conducted a statistical analysis. For each participant, statistics (mean M, standard deviation SD) were computed on the 18 trials in each condition. A one-way within subject design ANOVA (Techniques: Barrier Tape, Freeze-backup, 2:1-Turn) on the mean completion time (in seconds) revealed a significant main effect of the technique ($F(2, 22) = 183.22$, $p < 0.001$) (Figure 6.7). Follow up t tests revealed that completion time in the Barrier Tape technique ($M = 6.37$ sec, $SD = 1.30$ sec) was significantly shorter than in the Freeze-backup technique ($M = 21.49$ sec, $SD = 3.11$ sec, $t(11) = -19.15$, $p < 0.001$). Similarly, completion time in the Barrier Tape technique was significantly shorter than in the 2:1-Turn technique ($M = 14.54$ sec, $SD = 2.41$ sec, $t(11) = -14.61$, $p < 0.001$).

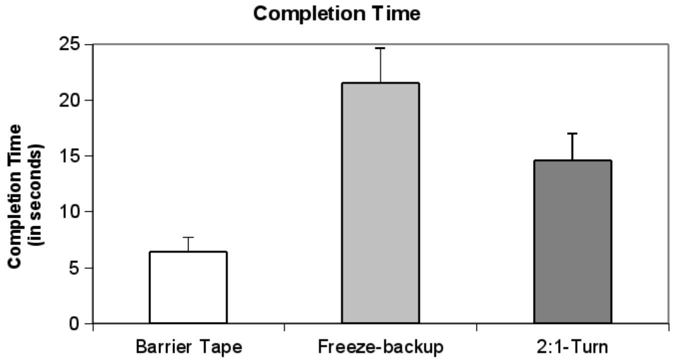


Figure 6.7 – Experiment #1: mean and standard deviation of the completion time (in seconds) for the three techniques (Barrier Tape, Freeze-backup, and 2:1-Turn).

6.3.1.2.b Amplitude of Walking in the Real World.

An ANOVA on the mean amplitude of walking in the real world (in meters) revealed a significant main effect of the technique ($F(2, 22) = 434.75$, $p < 0.001$). Follow up t tests revealed that the amplitude of walking in the real world in the Barrier Tape technique ($M = 1.46$ m, $SD = 0.16$ m) was significantly shorter than in the Freeze-backup technique ($M = 4.42$ m, $SD = 0.30$ m, $t(11) = -30.13$, $p < 0.001$). Similarly, the amplitude of walking in the real world in the Barrier Tape technique was significantly shorter than in the 2:1-Turn technique ($M = 3.37$ m, $SD = 0.23$ m, $t(11) = -20.80$, $p < 0.001$).

6.3.2 Experiment #2: Path Following Task

In the second experiment, our goal was to compare the 3 techniques over a path following task where the user had to follow a path delimited by two virtual walls, as fast as possible and as accurately as possible by trying to stay right between the two walls. We a priori assumed that the Magic Barrier Tape will be faster, as in Experiment #1, but less precise due to the controllability of rate control [54].

6.3.2.1 Description

6.3.2.1.a Population.

The population that participated in this experiment was the same as for Experiment #1.

6.3.2.1.b Experimental Apparatus.

We used the same experimental apparatus as in Experiment #1, except that we replaced the targets by two possible paths: a 2.5m radius circle or a 6m side length square (Figure 6.8). The walls at both sides of the path were in a semi-transparent blue material, 1m high, and at 1m from the path, creating a 2m wide corridor. Red arrows on the floor indicated the direction to follow, and a red 1m high and 0.2m radius cylinder indicated the start and finish position. The paths alternated throughout the experiment. Figure 6.9 shows the VE with the circular path as seen from the user's point of view.

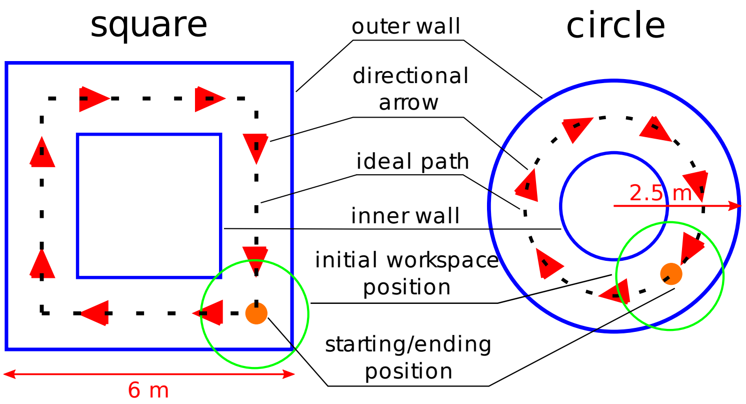


Figure 6.8 – Two paths used in Experiment #2, in top-view, with the initial cylindrical walking workspace position.

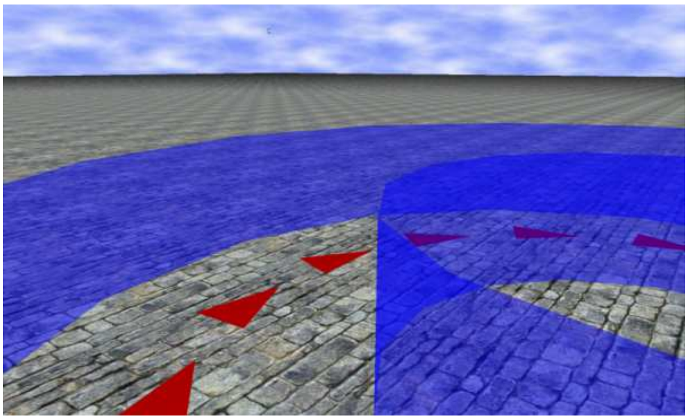


Figure 6.9 – The VE used in the Experiment #2 with circular path, walls and directional arrow cues on the ground.

6.3.2.1.c Procedure.

Before each trial, the subject had to go back to the initial position and orientation at the center of the workspace. Then, the experimenter launched the next trial. The participant was instructed to try to follow the path right between the two walls in the direction given by the arrows on the floor, until he reached the target cylinder. For each technique, he had 2 trials for training. The experiment lasted about 30 minutes, including training trials.

6.3.2.1.d Experimental Plan.

Participants completed all the three technique conditions (Barrier Tape, Freeze-backup, 2:1-Turn) and the order of the conditions was counterbalanced across participants. In each condition, the participants were exposed to 2 successive blocks of 2 trials with 2 different paths (square and circle). In each block, the presentation order of these trials was randomized. Participants completed a total of 12 trials (2 paths × 3 technique conditions × 2 trials per condition). During a learning phase, prior to each technique condition, participants were exposed to 2 trials that did not enter in the final data set.

6.3.2.1.e Collected Data.

Along with the same data as in Experiment #1, we also collected the path deviation. The path deviation (in m^2) is given by the area delimited by the subject's path in the VE and the ideal path (exactly between the two walls).

6.3.2.2 Results

For the different comparison analysis, a correction for experiment-wise error was realized by using Bonferroni-adjusted alpha level ($p = 0.05$ divided by the number of tests). Thus, in order to compare the Barrier Tape technique to the two other techniques (Freeze-backup and 2:1-Turn) the alpha level was adjusted to $p = 0.025$.

6.3.2.2.a Completion Time.

An ANOVA on the mean completion time (in seconds) revealed a significant main effect of the technique ($F(2, 22) = 84.01$, $p < 0.001$) (Figure 6.10). Follow up t tests revealed that completion time in the Barrier Tape technique ($M = 31.62$ sec, $SD = 9.71$ sec) was significantly shorter than in the Freeze-backup technique ($M = 99.54$ sec, $SD = 21.63$ sec, $t(11) = -12.06$, $p < 0.001$). Similarly, completion time in the Barrier Tape technique was significantly shorter than in the 2:1-Turn technique ($M = 52.33$ sec, $SD = 6.59$ sec, $t(11) = -6.48$, $p < 0.001$).

6.3.2.2.b Path Deviation.

An ANOVA on the mean path deviation (in square meters) revealed a significant main effect of the technique $F(2, 22) = 4.77$, $p = 0.019$ (Figure 6.10). Follow up t tests revealed that the path deviation in the Barrier Tape technique ($M = 3.46\ m^2$, $SD = 1.76\ m^2$) was not significantly different from the path deviation in the Freeze-backup technique ($M = 2.45\ m^2$, $SD = 1.04\ m^2$, $t(11) = 1.72$, $p = 0.11$). By contrast, the analysis indicated that the path deviation in the 2:1-Turn technique ($M = 1.93\ m^2$, $SD = 0.54\ m^2$) was significantly lower than in the Barrier Tape technique, $t(11) = 2.81$, $p = 0.017$.

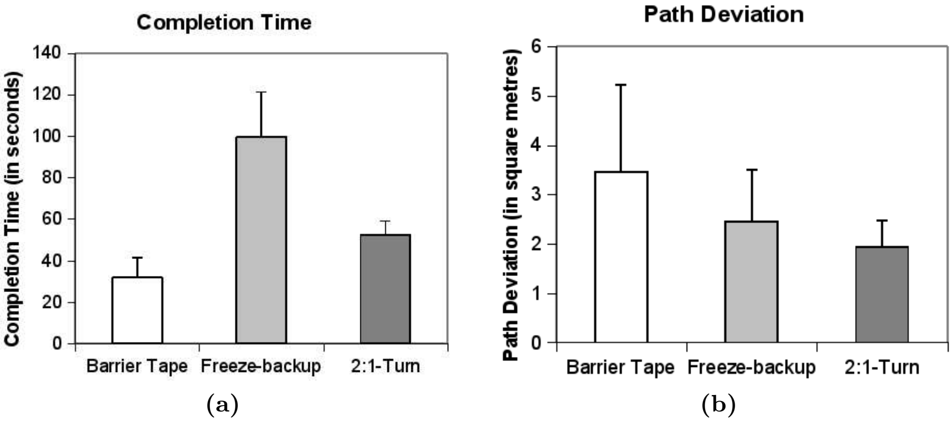


Figure 6.10 – Experiment #2: means and standard deviations of the completion time (a) and the path deviation (b) for the three techniques (Barrier Tape, Freeze-backup, and 2:1-Turn).

6.3.2.2.c Amplitude of Walking in the Real World.

An ANOVA on the mean amplitude of walking in the real world (in meters) revealed a significant main effect of the technique ($F(2, 22) = 379.81$, $p < 0.001$). Follow up t tests revealed that the amplitude of walking in the real world in the Barrier Tape technique ($M = 6.81$ m, $SD = 1.33$ m) was significantly shorter than in the Freeze-backup technique ($M = 19.03$ m, $SD = 1.25$ m, $t(11) = -32.63$, $p < 0.001$). Similarly, the amplitude of walking in the real world in the Barrier Tape technique was significantly shorter than in the 2:1-Turn technique ($M = 13.61$ m, $SD = 1.54$ m, $t(11) = -13.17$, $p < 0.001$).

6.3.3 Subjective Questionnaire

After both experiments, a preference questionnaire was proposed in which participants had to grade from 1 to 7 the 3 techniques according to 6 subjective criteria: *easiness of use*, *fatigue*, *navigation speed*, *navigation precision*, *general appreciation* and *naturalness*. Figure 6.11 shows the means and standard deviations of the 3 techniques for each of the subjective criteria.

Wilcoxon signed rank tests with Bonferroni correction showed significant differences: for the *fatigue*, between the Barrier Tape and the Freeze-backup techniques ($z = 2.69$, $p = 0.007$) and between the Barrier Tape and the 2:1-Turn techniques ($z = 2.41$, $p = 0.016$); for the *naturalness*, between the Barrier Tape and the Freeze-backup techniques ($z = 2.77$, $p = 0.006$) and between the Barrier Tape and the 2:1-Turn techniques ($z = 2.53$, $p = 0.011$); for the *navigation speed*, only between the Barrier Tape and the Freeze-backup techniques ($z = 2.82$, $p = 0.005$); and for the *general appreciation*, only between the Barrier Tape and the Freeze-backup techniques ($z = 2.65$, $p = 0.008$).

6.4 General Discussion

Both experiments showed that the Magic Barrier Tape is faster compared to the other techniques. Indeed, results show that Experiment #1 trials were completed more than 3 times faster with the Magic Barrier Tape than with the Freeze-backup technique, and more than 2 times faster than with the 2:1-Turn technique. In Experiment #2, completion

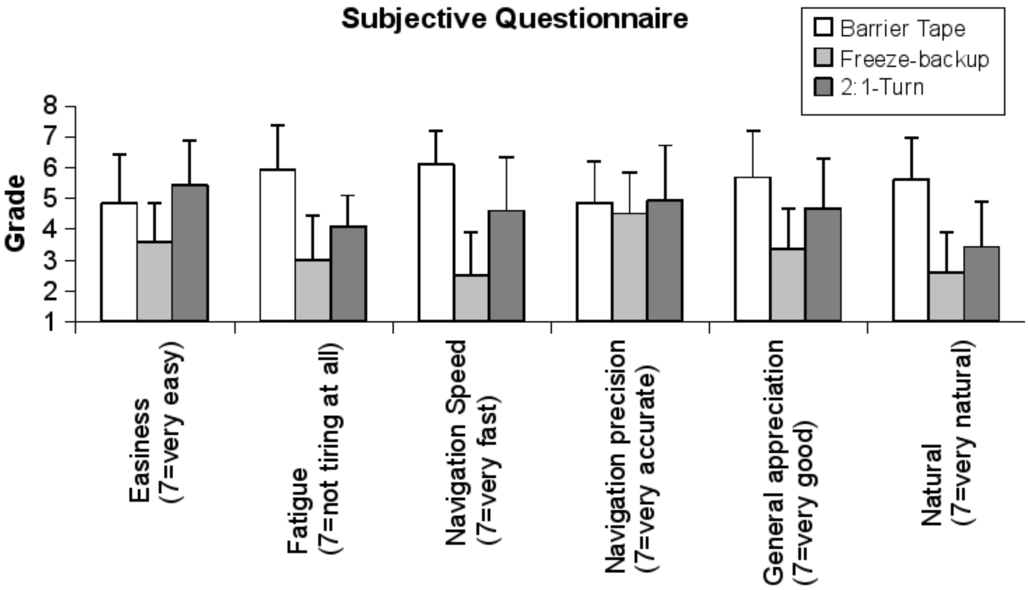


Figure 6.11 – Mean and standard deviation of subjective ratings about the different criteria for the three techniques.

time using the Magic Barrier Tape was also roughly 3 and 2 times faster respectively. This result is consistent with the user's impression from the questionnaire regarding the navigation speed of the different techniques (Figure 6.11). It is mainly due to the fact that there is no time lost in the resetting of the position when using the Magic Barrier Tape, and that the control law allows navigation speeds greater than the average walking speed. Completion times could be further reduced by tuning the control law for greater speeds, although controlling the Magic Barrier Tape could become increasingly difficult, as testified by 3 users which complained about an acceleration behavior that was sometimes hard to control.

The experiments also showed that users walked less when using the Magic Barrier Tape than with the other 2 techniques, which was expected due to the use of rate control at the boundaries of the workspace. However, an interesting observation can be made when considering that trials were completed significantly faster with the Magic Barrier Tape. If we do a ratio between the amplitude of walking in the real world and completion time, in a per user basis, we obtain similar values for the Magic Barrier Tape ($M = 0.24$, $SD = 0.04$), the Freeze-backup ($M = 0.21$, $SD = 0.04$) and the 2:1-Turn ($M = 0.24$, $SD = 0.04$) techniques in Experiment #1, as well as in Experiment #2 with ($M = 0.22$, $SD = 0.036$), ($M = 0.20$, $SD = 0.037$) and ($M = 0.26$, $SD = 0.052$) respectively. Hence, the amount of "useful walking", contributing to moving forward in the VE, relative to time is as large with the Magic Barrier Tape as with the other techniques. If we consider that walking speeds are the same for the 3 techniques, users spend roughly the same percentage of the total time doing useful walking with the Magic Barrier Tape technique as with the other 2 techniques.

Experiment #2 showed that the Magic Barrier Tape was less precise when following a given path, with a higher path deviation when compared to the 2:1-Turn technique (roughly 2 times less precise). We cannot conclude on the comparison with the Freeze-backup technique, since results were not significantly different. Again, these results were expected. By nature and design, the use of the Magic Barrier Tape is meant for coarse positioning. The user gets close enough to the navigation target in order to have it inside his workspace, and can then reach it by fine positioning navigation through natural walking. As explained by one of the subjects of the Experiment #2, when asked about the strategies he used: “I sent the barrier tape as far as possible without going into the walls in order to take advantage of the workspace”. However, path deviation could be improved by allowing users to cutomize their control law, like when they choose the mouse speed in desktop computers. In addition, Zhai [54] observed that, with sufficient training, rate control and position control can achieve similar performances. Hence, further user training on the Magic Barrier Tape rate control might improve its mean path deviation.

Overall, users graded the Magic Barrier Tape higher in all criteria of the questionnaire where comparisons were significantly different. We highlight that 6 subjects complained about having cybersickness when using the 2:1-Turn technique, while 2 said it made them loose balance. Many subjects found the Freeze-backup technique exhausting and frustrating. It is also important to note that 2 subjects had a very hard time using the Magic Barrier Tape. They used an inadequate strategy, and complained about the control law. They might have needed a longer training time, or more guidance on the strategy to adopt. They consistently graded it lower than the other techniques in every criteria of the questionnaire.

In a nutshell, the Magic Barrier Tape is faster than the Freeze-backup and the 2:1-Turn techniques, but is less precise when using it in rate control. The 2:1-Turn technique is the most precise, but seems to induce cybersickness to users, as well as stability issues. There is a general dissatisfaction with the Freeze-backup technique, mainly due to its physical exertion and slow speed, leading to a frustrating experience. People generally prefer the Magic Barrier Tape, and find it more natural and less tiring.

6.5 Conclusion

Navigating in large VE within the confines of a VR setup with limited translation space is inherently problematic. Users eventually reach the boundaries of the workspace, raising potential collision issues, tracking loss, and breaks of immersion. In this chapter, we address this problem by introducing the Magic Barrier Tape, a new interaction metaphor for navigating in a potentially infinite VE while confined to a restricted walking workspace.

We leverage the barrier tape metaphor and its “do not cross” implicit message by surrounding the walking workspace with virtual barrier tape in the VE. The technique uses a hybrid position/rate control mechanism: natural walking is used inside the workspace, while rate control navigation is used to move beyond the boundaries by “pushing” on the virtual barrier tape. Thus, the technique naturally informs the user about the boundaries of his walking workspace, providing a walking environment safe from collisions and tracking problems.

We conducted two experiments in order to evaluate the Magic Barrier Tape by comparing it to other state of the art navigation techniques addressing the same issues, previously extended for omni-directional navigation. In Experiment #1 participants had to walk to a target, while in Experiment #2 they had to follow a path. Results showed that the Magic Barrier Tape was faster than the other techniques. Experiment #2 results confirmed that, by design, navigation through rate control with the Magic Barrier Tape is not meant for precise path following, but rather for coarse positioning between fine positioning tasks. Overall, the Magic Barrier Tape was more appreciated, while being more natural and less tiring.

Our approach can be potentially used in different VR setups where reaching the boundaries when walking is a problem. However, some VR setups have additional boundary constraints if the visual display does not entirely surround the user. This is the case for CAVE-like setups, where there is one or several missing screens. These constraints in rotation arise when the user turns and faces the missing screens, thus breaking user immersion. In the next chapter, we address this additional constraint by extending the Magic Barrier Tape and designing 2 other novel techniques.

7

Infinite Navigation in Large Virtual Environments within Restricted Translation and Rotation Workspaces

Contents

7.1 Three Novel Navigation Techniques 128
  7.1.1 Motivation for New Navigation Metaphors 129
  7.1.2 General Terminology and Quantities 129
  7.1.3 Constrained Wand and Signs 130
  7.1.4 Extended Magic Barrier Tape 131
  7.1.5 Virtual Companion 132
7.2 Evaluation 133
  7.2.1 Experimental Conditions 134
  7.2.2 Task #1: Pointing Task 135
  7.2.3 Task #2: Path Following Task 136
7.3 Results 136
  7.3.1 Recorded time and tracking data 136
  7.3.2 Questionnaire 138
7.4 Discussion 139
7.5 Conclusion 141

Immersive spaces such as 4-sided (cube or CAVE-like) displays with stereo viewing and high-quality tracking provide a very engaging and realistic virtual experience. Recent advances in tracking and display technologies have now made it possible to immerse users in complex and engaging VE, resulting in a very strong feeling of presence.

However, such setups are inherently limited by the restricted physical space. In the previous chapter, we introduced a novel metaphor allowing the navigation of infinite VE within the confines of a restricted size workspace, where users can only walk for a few steps before encountering the physical screens (limited translation). However, there is often one or several missing screens on CAVE-like setups: when the activity in the VE requires the user to turn, he might face a missing screen (limited rotation) thus breaking the presence illusion.

In this chapter, we propose three new techniques (the *Constrained Wand and Signs*, the *extended Magic Barrier Tape*, and the *Virtual Companion*) which address translation *and* rotation boundaries in different ways and to different degrees. First, they all incorporate a *warning* technique which prevents users from hitting the walls or seeing the missing screen.

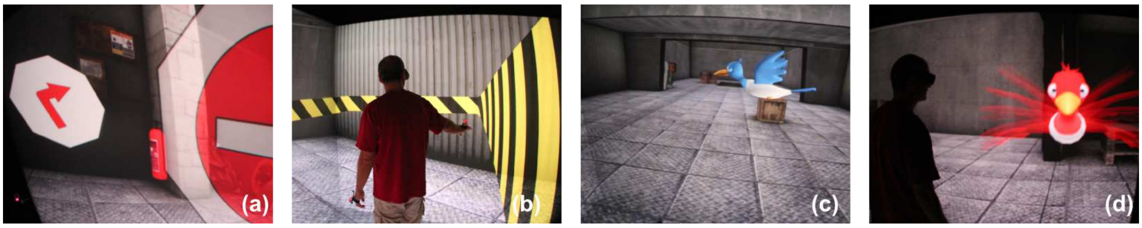


Figure 7.1 – Screenshots illustrating the three techniques. From left to right: (a) Constrained wand and signs: The "no-way" and "turn right" signs. (b) Extended MBT: the tape and blinders. (c,d) Virtual Companion: the bird in "rest mode" (c) and "protection mode" (d).

Second, they incorporate a *navigation* technique which enables displacement to out-of-reach locations using a rate-control paradigm. These three techniques were developed in a continuum, from less to more ecological: each technique is progressively more integrated with the VE. We designed and ran a user study to compare the relative advantages of each approach, and compared them to a traditional wand navigation.

This chapter is organized as follows. We present the objectives and motivations for the different metaphors in Section 7.1, as well as a detailed presentation of each technique. The techniques are evaluated following an experiment described in Section 7.2, using a wand "baseline" condition and pointing and navigation tasks. The results of the experiment are presented in Section 7.3, and are discussed in Section 7.4, where we provide insights into the various trade-offs involved in these navigation metaphors.

7.1 Three Novel Navigation Techniques

Extending the objectives of Chapter 6, our goal is to address the limitations of CAVE-like immersive displays, namely 1) keep the user "safe" from reaching translational and rotational limits, and 2) provide more enjoyable and ecological paradigms compared to other navigation techniques, such as traditional wand-based navigation techniques. One of the conclusions of the study of the Magic Barrier Tape from the previous chapter was that the technique made users walk less than resetting techniques. Since we are basing our approach on natural walking navigation due to its immersive and performance properties [23, 20, 21], as stated in the introduction of this manuscript, we add a third objective: 3) increase the amount of real walking compared to traditional techniques.

At first glance, these goals may seem simple or just plain common sense. However, developing navigation metaphors achieving these goals can be surprisingly hard. There has been a significant body of work on walking in VE, as shown in Chapter 5. Most notably, for CAVE-like spaces, techniques such as "Walking in Place" [181] have been proposed; however they do not involve true physical walking. Several approaches have been developed for modifying the walking path (e.g., "Redirected walking" [36]), but they usually require a space which is larger than a typical cube-like display. The few that address the rotation boundary issue do not provide a convincing solution.

In this section, we present our three new navigation metaphors designed to achieve these goals.

7.1.1 Motivation for New Navigation Metaphors

Our three novel metaphors can be seen as varying from simple (and trivial) to more complex, and from less to more ecological. The techniques are indeed progressively more embedded in the VE. We wanted to avoid using rotation and translation scaling due to their noticeability in small VE [36, 187] and their need of specific VE. Instead, we used hybrid position/rate control approaches, based on the good results of the Magic Barrier Tape.

The design of the three metaphors follows a similar methodology which could be re-used later for conceiving new and other techniques. All the metaphors incorporate two main components: (1) a warning technique and (2) a navigation technique. The "warning" technique is meant to prevent the users from colliding with the walls or perceiving the missing screen. As such it satisfies our safety criterion (first goal). The "navigation" technique is mandatory when facing a restricted workspace. It is meant to extend the walking capabilities to out-of-reach locations.

To do so, different implementations of a rate-control paradigm are used corresponding to the different metaphors used: from a standard wand-based controller to a more elaborate gesture-based paradigm (third goal). Besides, for two of the techniques the rate-control mode is not possible when the user is located far from the boundaries of the workspace, i.e., inside the central (and thus safer) area. This satisfies our second goal by strongly encouraging the user to walk whenever they are located inside the safe area of the restricted workspace.

The first technique method is a relatively straightforward extension of a wand metaphor: the user is forced to walk to the cube limits before the wand is activated. At the limits, warning signs are presented before collisions (translation) or when turning the head too far (rotation). At this point the wand can be used as usual. This metaphor both encourages walking and keeps the user safe (Fig. 7.1(a), 7.3(left)).

The second technique extends the Magic Barrier Tape, presented in Chapter 6, protecting the user with the tape for translation, and introducing "virtual blinders" for rotation. The user can move beyond the physical space by pushing the barrier with his hands (Fig. 7.1(b)). By construction, the barrier only appears when approaching the walls or rotating too far. As a result this metaphor also encourages walking, keeps the user safe and removes the need for the wand.

The third novel metaphor is the *Virtual Companion*, which is an animated bird in the VE. The bird serves two purposes: it protects the user at the limits by becoming red and flying close to the user's face (Fig. 7.1(d)), and can be "tethered" with virtual reins, thus serving as a navigation interface. This approach protects the user, leaves their hands free and is the most ecological of the three. However, it only slightly encourages walking, by forcing users to step backwards when they are too close to the screens or have turned too far.

7.1.2 General Terminology and Quantities

Before presenting the three techniques in detail, we introduce some terminology and the corresponding boundaries in the physical workspace. In our case, this is a 4-sided cube-shaped space (3 walls and the floor) measuring 3.2 m wide × 3.2 m long × 2.4 m high.

At a high level, we define a "safety zone" in which the user is not looking at the missing screen and is not close to the walls of the cube. We then define a "danger zone" in which

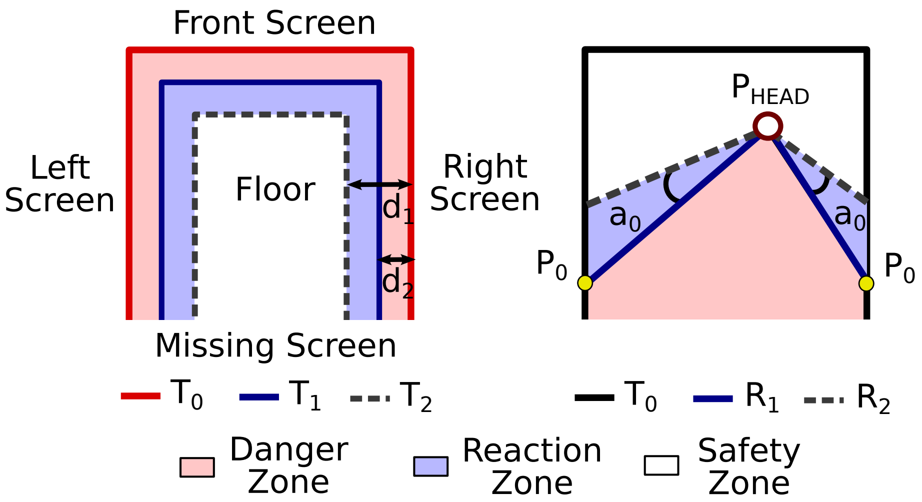


Figure 7.2 – Regions and boundaries for translation (left) and rotation (right) common to all techniques.

there is a danger of physical collision with the walls or of seeing the missing screen, and a "reaction" zone in which boundary cues are progressively presented, warning the user that they are getting close to the danger zone. These zones are common to all techniques.

For translation, these zones are defined by the following translation boundaries, illustrated in Fig. 7.2(left).

- T_0 is the physical position of the walls.
- T_1 bounds the danger zone for translations, with $T_1 = T_0 - d_1$. The translation danger zone is between T_1 and T_0, where there is a risk of collision with the physical walls; we set $d_1 = 40$ cm for our setup, empirically leaving enough space for comfortable arm motion.
- T_2 bounds the reaction zone, with $T_2 = T_0 - d_2$. The translation reaction zone is between T_2 and T_1, while the translation safety zone is between the workspace center and T_2. We set $d_2 = 75$ cm, which empirically allows ample warning for reasonably fast motion.

For rotation, these zones are defined as follows and illustrated in Fig. 7.2(right):

- Vector R_1 bounds the danger zone for rotations. It is defined by the vector $P_0 - P_{head}$, where P_0 is a position on the side walls, and P_{head} is the position of the user's head. P_0 is at a small distance from the missing screen. When turning beyond R_1, the user is in the rotation danger zone. The angle between the head orientation and R_1 defines the angular distance from the user to the rotation danger zone.
- Vector R_2 bounds the reaction zone for rotations. It is defined by the angle a_0 to R_1. The rotation reaction zone is between R_2 and R_1, and the rotation safety zone is before R_2. We set $a_0 = 25°$ considering a reasonably fast rotation motion of a user.

7.1.3 Constrained Wand and Signs

Concept. The first technique, Constrained Wand and Signs (CWand) encourages users to walk by restricting the zone in which the wand can be used. Protection is provided

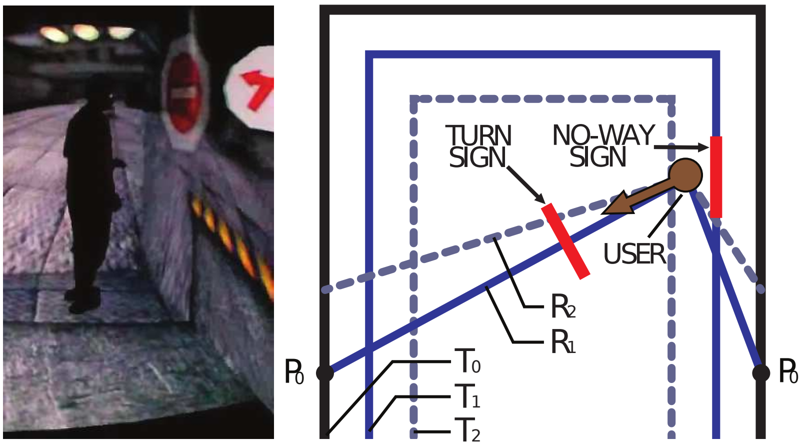


Figure 7.3 – The constrained wand: signs become visible to warn for rotation and translation.

with warning signs which appear when the user is in the reaction and danger zones. See Fig. 7.3 for a schematic illustration.

Warning. A semi-transparent "no-way" sign appears when the user reaches T_2 and becomes fully visible at T_1. For rotation, a "turn-left" sign appears when the user turns right into the reaction zone (equivalently "turn-right"). Fig. 7.1(a) shows both visuals.

Navigation. Users can use the wand only in the reaction and danger zones. In these zones, the wand is operated in the customary manner. The forward translation direction is given by the wand direction.

Implementation. For rotations, the sign position is in the viewing direction and in the reaction zone its distance varies linearly between 0.7 m and 1.2 m from the user's head position.

7.1.4 Extended Magic Barrier Tape

Concept. The Extended Magic Barrier Tape (eMBT) is an extension of the Magic Barrier Tape to the more limited cubic workspace, in particular by adding warnings for the rotational limit due to the missing screen. Walking is encouraged since navigation is only possible in the reaction and danger zones, while protection is provided with the tape and the new "blinders" for rotation.

Warning. The "translation tape" is located at T_1 and is aligned with the walls. It appears semi-transparently when the user is at T_2 and becomes fully visible at T_1. For rotation, we have introduced "virtual blinders". These start at the side of the translation tape (see Fig. 7.4(right)), and progressively become wider until they completely block the view of the user when reaching the head (see Fig. 7.1 (b)).

Navigation. The user translates by pushing the tape forward with a tracked hand. The speed depends linearly on how far the user pushes, and the direction is defined by the positions of the head and the hand. Rotation is achieved by a "pushing back" gesture on

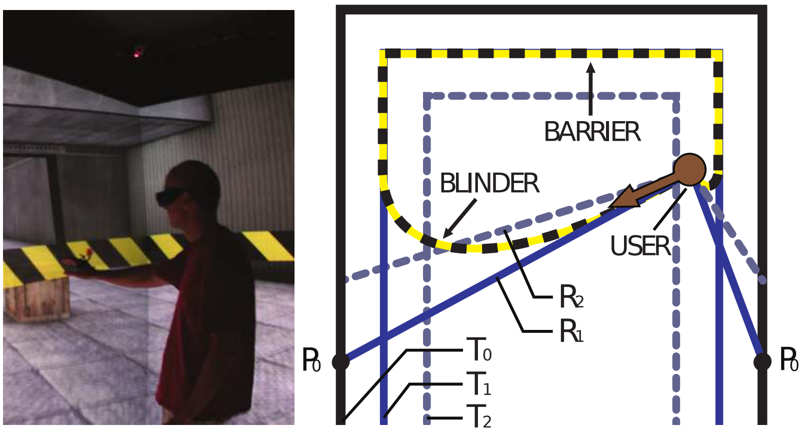


Figure 7.4 – The extended MBT: the tape follows the walls, and blinders warn for rotation.

the blinders.

Implementation. We experimentally set the height ("altitude") of the tape to 1.4 m; it is infinitely thin, has a width of 15 cm and a yellow/black slanted texture imitating real barrier tape. The blinders follow Bézier curves with the tape/blinder intersection, P_0 and the user head position as control points. The width of the blinder at the user's head is 60 cm.

7.1.5 Virtual Companion

Concept. The Virtual Companion (VC) is the most innovative of the three techniques we introduce. An interactive "companion" is used to implement warning and navigation (see Fig. 7.1(c)-(d)). In our implementation we used a bird with flapping wings, but other representations could be used (dog, fairy etc.). In contrast to the previous techniques, navigation can be achieved in the safety zone. However, if the user is in the danger zone, they must step back into the safety zone to initiate navigation.

Warning. When the user is in the safety zone, a blue bird stays near the closest wall, calmly flaps its wings and hovers at the height of the head (see Fig. 7.5, 7.1(c)). We call this "rest" mode. In the translation reaction zone, the bird's position is at the position of the user's head projected onto the closest wall. We call this "following mode". In the danger zone (translation and rotation), the bird flies directly in front of the user's face, turns red and angrily flaps its wings (Fig. 7.1(d)). This is "protection mode".

Navigation. We introduce a set of gestures to navigate with the VC using virtual reins. These are illustrated in Fig. 7.6. The user invokes "navigation mode" by bringing their hands together for 1 second. The reins then appear, attaching the user's hands to the VC. The VC moves in front of the user. To cancel navigation mode, the user crosses their arms. To move forward the user imitates a move forward command that would be given to a horse (moving the reins up and down). Once the motion has started, the user moves their hands forward to accelerate, and pulls back to decelerate and eventually stop. Rotation is achieved by moving the corresponding hand to the side. The rotation speed is given by the distance between the hands. To stop turning, the active hand is moved back next to the other hand. The implemented gestures are given as examples but future work could

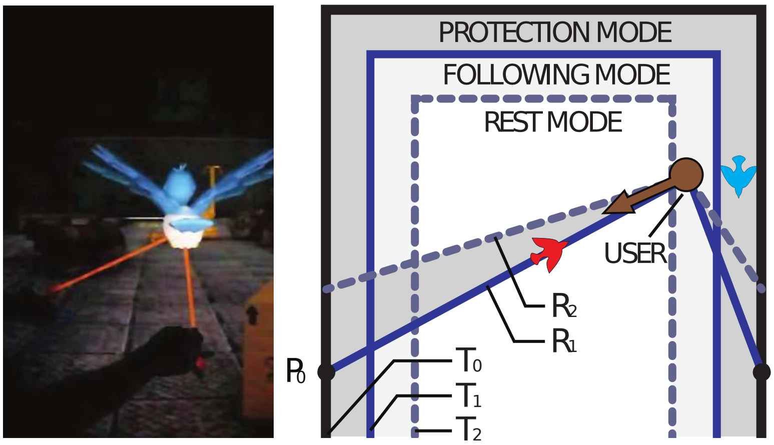


Figure 7.5 – The Virtual Companion. Left: the gesture for turning. Right: the bird changes position and color to warn for rotation and translation.

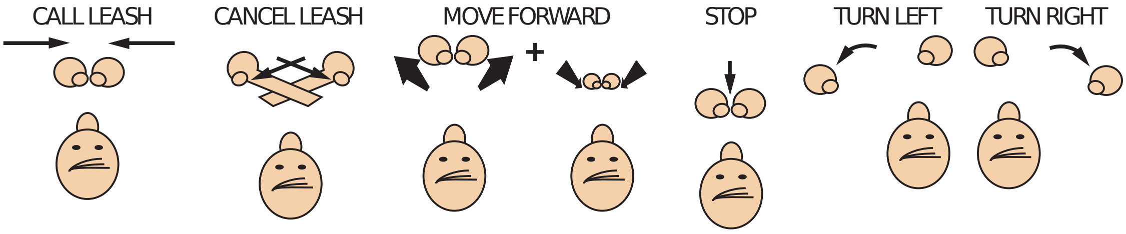


Figure 7.6 – The gesture set for controlling the Virtual Companion.

focus on the best way to control the Virtual Companion.

Implementation. The bird follows Bézier curves using the current position and past and future goal positions for smooth interpolation between trajectories. Rotation protection has priority over the translation protection, except when reaching T_1 where priorities are inverted. Activating navigation mode is the most sensitive gesture: the user has 1 s to complete an up-down movement of 25 cm with both hands.

7.2 Evaluation

To evaluate the different metaphors described above, we compare the three new navigation techniques, namely Constrained Wand, Extended Magic Barrier Tape, and Virtual Companion. We also include in the comparison the "baseline condition" of a standard Wand interface, typically used for navigation in CAVE-like setups, which is expected to be faster and possibly more accurate. The only enhancement for the wand is the appearance of warning signs for collision, thus avoiding physically bumping into walls. We chose this approach to provide a fair comparison; in typical workspaces the experimenter/operator uses a verbal warning to avoid collisions which adds a modality unavailable in the other techniques. We do not compare them to resetting techniques as in Chapter 6, since resetting techniques do not deal with rotation boundaries, and the Magic Barrier Tape was shown to be more efficient and more appreciated.

We conducted two experiments corresponding to two different tasks, a pointing task and a path following task. Both tasks were performed first in a simple VE with minimal

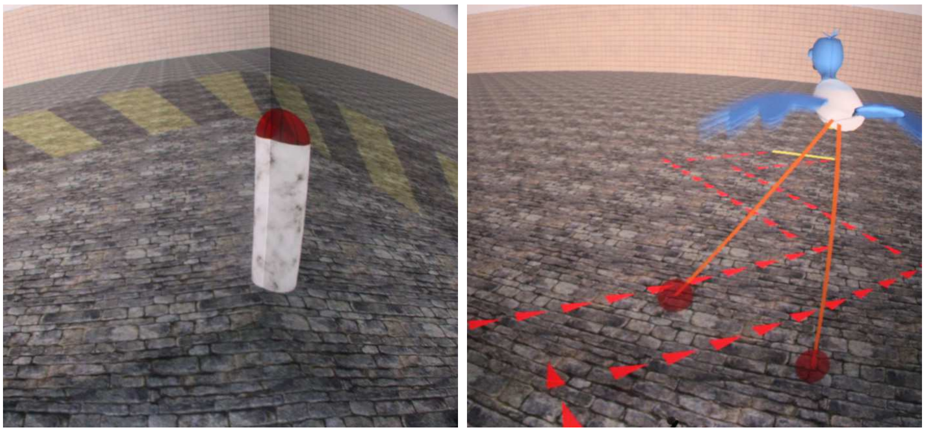


Figure 7.7 – The simple VE used for the first block of tests in our study.Left: simple target object, right: zigzag path.

visual cues to distance and orientation (Fig. 7.7), and later in a complex VE with more realistic visual cues (Fig. 7.1). The aim of the pointing task was to evaluate the walking distances during the navigation in the VE. We expected that the Extended Magic Barrier Tape and Constrained Wand will encourage users to walk more than the other techniques. The aim of the path following task was to test the navigation in all directions, including the missing screen. We expected that the Virtual Companion will help users to stay in the safety zone.

7.2.1 Experimental Conditions

7.2.1.1 Population

Twelve participants (3 females and 9 males) aged from 23 to 61 ($M = 30.7, SD = 10.0$) took part in this experiment. Three of them were left-handed. Three of them wore glasses or contact lenses to correct for myopia. None had any other known vision or perception disorders. They were all unpaid volunteers and naïve to the purpose of the experiment.

7.2.1.2 Experimental Apparatus

The experiment was conducted in a 4-screen cube (3 walls and the floor) measuring 3.2 m wide × 3.2 m long×2.4 m high. Each screen uses Infitec stereo running at 60 Hz per eye, and was driven by a separate NVIDIA Quadro 5800 GPU to ensure a constant 60 · fps per eye. The position and orientation of the user's head and hands and of the wand were tracked by an ART 6-camera infrared tracking system running at 60 Hz.

Two VE of different complexity were constructed for the purpose of these experiments. The simple VE consisted of a large square room (50 m×50 m) with a stone floor texture, tiled walls, and an untextured ceiling (see Fig. 7.7). The complex VE was an industrial hangar with a central open area where the targets were placed (Fig. 7.1). Outside the experiment area, it was cluttered with machines, crates, and barrels, and two more rooms were visible. This VE had realistic distances and sizes, and high resolution textures with baked global illumination.

For pointing tasks, a target was placed in the VE. In both VE, the target was a marble cylinder with a red hemispherical cap resembling a button on top. For path following tasks, the edges of the path were marked on the floor by red arrows indicating the direction to follow, and a yellow finish line indicated the end of the path. These can be seen in Fig. 7.7.

7.2.1.3 Procedure

Before the beginning of the experiment, the four techniques were explained to the subject in a training session taking approximately 15 min on average. The training VE consisted of the simple VE, in which a 3 m path with a single 90° corner was marked on the floor and three targets successively appeared. This training session accustomed participants to the workspace, the display technology, the navigation techniques, and the pointing and path following tasks.

In the experiment, participants first performed a block of pointing and path following tasks in the simple VE, and then a second block in the complex VE. Between the two blocks, there was a pause to fill in a questionnaire of 11 Likert items. This pause also served to avoid cybersickness from prolonged use of the workspace. After both blocks were completed, a second, subjective questionnaire of open questions concluded the experiment. Each block of trials lasted about 15 min, and the complete experiment, including training and questionnaires, lasted approximately 60 min on average.

7.2.1.4 Collected data

For each trial and each subject, we recorded the completion time (in seconds) and all the head tracker readings (i.e., position and orientation of the head). This data also allows the computation of the amplitude of physical walking (in meters). The amplitude of physical walking is the distance traveled when walking in the real world, which of course also causes an equal traveled distance in the VE. It also allows the computation of the path deviation from the ideal path, i.e., the shortest path to the target or the center line of the indicated path. The path deviation (in m^2) is given by the area delimited by the subject's path in the VE and the ideal path.

7.2.2 Task #1: Pointing Task

In Task #1, our goal was to compare the 4 techniques (3 new and Wand) in a pointing task where the user had to move from a central initial location to a new location, indicated by a target, as fast as possible. When the target is reached, the user touches the target with one hand and the task is completed.

Before each trial, the subject had to go back to the initial position and orientation. Then, the VE became visible and the participant was instructed to look for the target in the VE, move towards it, and touch it with one hand. The participant was instructed to use physical walking or virtual walking using the technique, or a combination of both, at their own choice. The target was always placed outside the limits of the physical workspace, so some amount of virtual walking was always required to complete the task.

Participants completed all four conditions (corresponding to the four navigation techniques) and the order of the conditions was counterbalanced across participants. In each condition, the participants were exposed to 2 trials, with the target either at 3.6 m straight in front of the user's initial position, or 4.4 m straight behind, alternating throughout the experiment. Participants completed a total of 8 trials (2 target positions × 4 technique conditions) in each of the 2 VE.

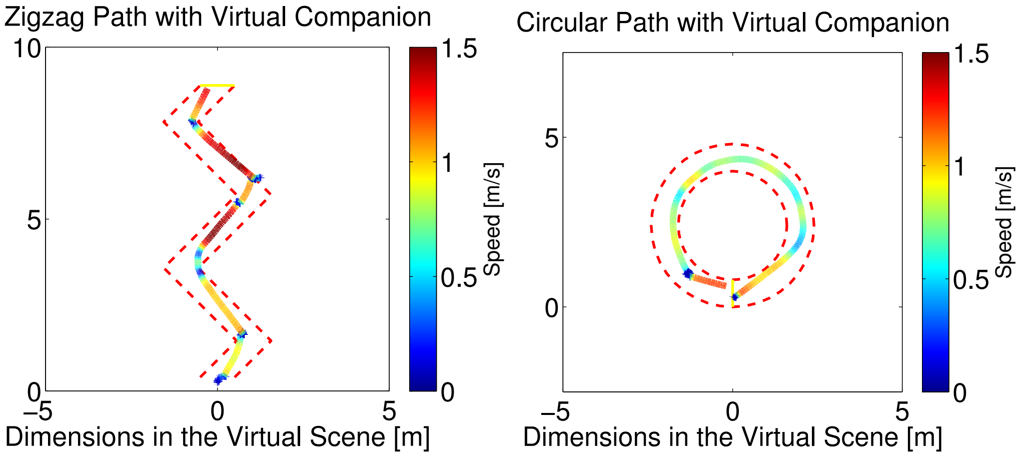


Figure 7.8 – Recorded trajectories of a participant performing (a) a zigzag path and (b) a circular path. The trajectories are the result of physical walking and/or Virtual Companion navigation and are color coded with the speed.

7.2.3 Task #2: Path Following Task

In Task #2, our goal was to compare the 4 techniques in a path following task where the user had to follow a path delimited by red arrows on the virtual ground (see Fig. 7.7(right)), as fast as possible and as accurately as possible by trying to stay in the middle of the path.

Before each trial, the subject had to go back to the initial position and orientation. Then, the VE became visible and the participant was instructed to follow the path in the direction indicated by the arrows on the floor, until they reached the finish line. Walking instructions were the same as the first task, and again some virtual walking was necessary since the path extended beyond the physical space.

As for Task #1, participants completed all conditions in counterbalanced order. In each condition, the participants were exposed to 2 trials, either with a zigzag path 12 m long and 70 cm wide, with four 90° corners (Fig. 7.8(a)), or with a circular path 12.5 m long and 80 cm wide (Fig. 7.8(b)), alternating throughout the experiment. Participants completed a total of 8 trials (2 paths × 4 technique conditions) in each of the 2 VE.

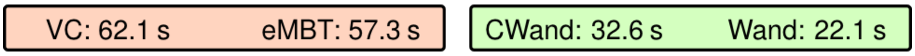
7.3 Results

7.3.1 Recorded time and tracking data

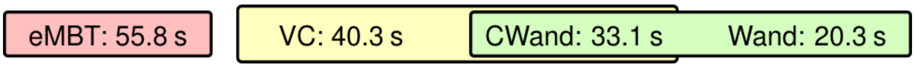
One-way ANOVAs were used to test for differences among the 4 navigation techniques according to several criteria. Tukey-Kramer post-hoc comparisons were performed to determine the statistical equivalence groups of the 4 techniques for these criteria.

7.3.1.1 Time to complete the task

The time needed to complete the tasks only differed significantly across the 4 techniques for the zigzag path ($F(3,92) = 8902.49, p < 0.001$) and for the circular path ($F(3,92) = 5262.51, p < 0.001$). Figure 7.9(a)-(b) shows the statistical equivalence groups for these tasks. For the pointing tasks, all navigation techniques were equivalently fast, but in general the wands have a speed advantage for the precision navigation required in path



(a) Completion time, zigzag path



(b) Completion time, circle path

(c) Physical walking distance, target in front

(d) Physical walking distance, target behind

(e) Physical walking distance, zigzag path

(f) Physical walking distance, circle path

(g) Time spent in warning mode, target in front

Figure 7.9 – Equivalence groups of the 4 techniques for the recorded time and tracking data. (a)-(b) Time needed to complete the zigzag path and the circular path. The pointing tasks showed no significant effect of technique on the task completion time, i.e., all techniques were in a single equivalence group. (c)-(f) Physical walking distance for the 4 tasks. (g) Amount of time spent in reaction and danger zones during the pointing task with the target in front. All tasks had the same equivalence groups for reaction and danger zones, only with slightly different averages.

following tasks. The VC is faster than the eMBT on smoothly curved paths, but not on paths with sharp corners.

7.3.1.2 Physical walking distance

The amplitude of physical walking differed significantly across the 4 techniques for each task: target in front ($F(3,92) = 38.51, p < 0.001$), target behind ($F(3,92) = 54.70, p < 0.001$), zigzag path ($F(3,92) = 193.81, p < 0.001$), and circular path ($F(3,92) = 242.27, p < 0.001$). Figure 7.9(c)-(f) shows the statistical equivalence groups for these tasks. In general the eMBT and the CWand encourage physical walking because they can only be activated at the limits of the physical workspace. The VC and the Wand do not incite users to walk physically.

7.3.1.3 Time spent in reaction and danger zones

The amount of time spent in the zones where the warning visual cues are active, both in rotation and translation, differed significantly across the 4 techniques for each task: target in front ($F(3,92) = 6810.53, p < 0.001$), target behind ($F(3,92) = 13007.03, p < 0.001$), zigzag path ($F(3,92) = 16564.48, p < 0.001$), and circular path ($F(3,92) = 16299.34, p <$

0.001). Figure 7.9(g) shows the statistical equivalence groups for one representative task. Closer examination of the actions that trigger the warning cues reveals that the translation warning generally dominates this statistic. The CWand and the eMBT force users to enter the reaction zone closer to the limits of the physical workspace to activate the technique. Users almost never look in the direction of the missing screen while using the eMBT. Also, users almost never approach the limits of the physical workspace when using the VC, and its statistic is actually dominated by the rotation reaction and danger zones.

7.3.1.4 Deviation from the ideal path

The deviation from the ideal path only differed significantly across the 4 techniques for the pointing task with the target in front ($F(3, 92) = 1.95, p = 0.008$). However, the effect size is very small: the pointing task with the target in front was by far the most accurately performed task with an average deviation across all techniques, subjects, and VE of 0.65 m^2, while the next most accurate task, the zigzag path, had an average deviation of 7.15 m^2.

7.3.1.5 VE differences

One-way ANOVAs were used to test for differences between the two VE according to the same criteria. The only significant difference we found was a faster task completion time for the zigzag path in the hangar than in the simple VE. Since the hangar VE always followed the simple VE in the experiments, we cannot know whether this effect is due to learning or due to the increased visual cues. However, since no other significant differences were found and participants do find visually rich VE more enjoyable, we encourage the use of such VE for this type of navigation interface studies.

7.3.2 Questionnaire

After completing both tasks in the simple VE only, a preference questionnaire was proposed in which participants had to grade the four techniques from 1 (low) to 7 (high) according to 11 subjective criteria: (a) Accuracy, (b) Ease of use, (c) Fun, (d) Walking sensation, (e) Fatigue, (f) Speed, (g) Presence, (h) Cybersickness, (i) Safety (avoid bumping into walls), (j) Missing screen awareness, (k) Global appreciation. Figure 7.10 shows the results concerning the grades (Likert-scale) obtained by the four different techniques for 4 subjective criteria. By performing a Friedman test on the four different conditions, we found a significant effect for 8 criteria, including Accuracy ($\chi^2 = 9.89, p = 0.019$), Ease of use ($\chi^2 = 16.54, p < 0.001$), Fun ($\chi^2 = 19.36, p < 0.001$), Walking sensation ($\chi^2 = 16.34, p < 0.001$), and Missing screen awareness ($\chi^2 = 9.23, p = 0.03$). No significant effect was found for Cybersickness ($\chi 2 = 0.40, p = 0.94$); however, participants reported very low levels of cybersickness ($M = 1.73, SD = 1.41$).

Post-hoc analysis showed that the Wand was rated significantly higher than the eMBT for accuracy ($p = 0.009$) and ease of use ($p = 0.003$). The Wand was also rated significantly higher than the VC for ease of use ($p = 0.003$). The eMBT was rated significantly higher than the Wand for missing screen awareness ($p = 0.03$) and walking sensation ($p = 0.01$). The eMBT was also rated significantly higher than the VC for walking sensation ($p = 0.001$). Finally, the VC was rated significantly higher than the CWand for fun ($p = 0.007$). The VC was also rated significantly higher than the eMBT for fun ($p < 0.001$).

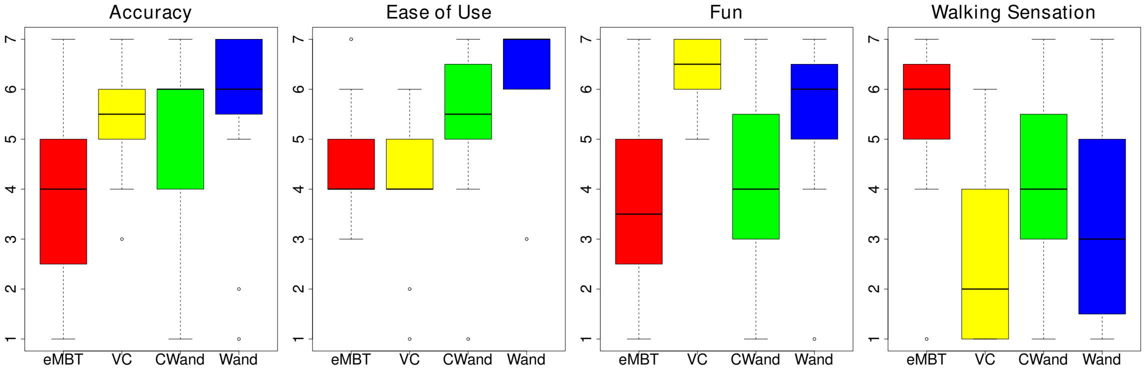


Figure 7.10 – Boxplots comparing the grades of the four techniques for some of the most interesting subjective criteria in the questionnaire.

7.4 Discussion

We proposed three novel interaction techniques for navigating in CAVE-like setup: an improved version of the Wand technique constraining the use of the wand to unsafe zones, an extended version of the Magic Barrier Tape adapted to CAVE-like setups and a completely novel metaphor called *Virtual Companion* to assist the user during his navigation. Our goals were: (1) to keep the user safe, (2) to encourage the user to walk and (3) to provide ecological interaction techniques.

Concerning the first goal, we see that the Virtual Companion outperforms all other techniques in terms of staying in the "safety zone" (Fig. 7.9(g)). Inevitably, this comes at the price of lower physical walking distance for all tasks except the zigzag path (Fig. 7.9(c)-(e)). However, in this case the additional walking was mainly due to position adjustments during navigation. Although participants spent more time in the protection zone for the eMBT compared to the VC, they gave higher ratings in the questionnaire to the criterion "missing screen awareness" compared to the Wand. Participants found the eMBT metaphor useful for warning them of the real workspace limits, as previously observed in the original version (Chapter 6).

For the second goal, the eMBT induces the largest amount of physical walking for three out of four tasks. This is also confirmed by the subjective questionnaire, where the technique was ranked higher than both the Wand and the VC in terms of walking sensation. In addition, both eMBT and CWand outperform the Wand in terms of physical walking distance (Fig. 7.9(c)-(e)). These observations meet our initial goal of proposing new metaphors allowing users to walk in restricted real workspaces. Our new techniques propose a nice alternative to previous techniques such as redirected walking [36] that can not always be deployed in CAVE-like setups.

For the third goal, evaluating the success of the ecological nature of the new metaphors is harder. The VC was judged to be significantly more fun than the CWand and the eMBT; interestingly there was no significant difference with the Wand. It must be noted however that the VC scored low on ease of use; learning the gestures did require some effort, and, like any gesture-based interface, tracking quality is paramount. In some cases tracking did deteriorate, causing some subject frustration especially for the start gesture. Nonetheless, both the eMBT and the VC succeed at providing a controller-free navigation metaphor, hence improving user interaction and freeing the hands for other tasks.

The presence question did not provide any significant results. However, in the free-

Criterion		**Wand**	**CWand**	**eMBT**	**VC**
Safety	translation	++	+	+	++
	rotation	- -	+	++	+
Physical walking		- -	+	++	- -
Ecological		- -	-	+	++
Fast navigation		++	+	-	-
Ease of use		++	+	-	-
Learning time		++	++	+	-
Fun		+	-	-	++
Requires wand		- -	-	++	++

Table 7.1 – Summary overview of the main advantages and drawbacks of our techniques.

form comments, 8 out of 12 subjects included positive remarks ("fun", "great" etc.) about the VC, while no such comments were given on the other techniques. The VC was also rated significantly higher than the eMBT and CWand for the "fun" criterion. We interpret this as an encouraging indication on approaches such as the VC. The "fun" aspect of the companion as the assistance tool provided to the participants instead of warning signs might represent a good alternative to existing techniques.

Nonetheless and as expected, the Wand technique is overall faster and more accurate, both in terms of objective measures (completion time, Fig. 7.9(a)-(b)), and from the subjective responses in the questionnaire (Sect. 7.3.2).

It is interesting to note that the goals of providing safety and encouraging walking can be contradictory: when using a wand, it is possible to go everywhere without moving or turning. Thus the safety goal is completely achieved but walking is not encouraged. We believe that this observation indicates a deeper issue which is confirmed by our results.

A second issue which follows from our results is that more complex –and in our case more ecological– interfaces come at the price of slower speed in task accomplishment. This was true both for eMBT and VC. To a lesser extent there is also an effect on task accuracy.

Both these issues are trade-offs which needs to be considered by application designers. Table 7.1 provides an overview of the different techniques we have proposed, summarizing the relative strengths and weaknesses of each. It could help developers choose the appropriate approach depending on their application needs. The different criteria reflect the main aspects that were evaluated and discussed in this chapter. This overview takes into account statistical data as well as subjective feedback and observations made during the experiment. As such it should be considered cautiously. Still, it seems that if speed and accuracy are paramount, more traditional controller interfaces may still be appropriate, but augmented with our novel signs and warning techniques for improved safety. If physical walking in the VE is more important, using a paradigm such as eMBT or CWand should be preferred. Last, fun and ecological criteria would favor the VC and its gestural interaction with a friendly virtual character.

Of course, an ideal navigation interface for restricted spaces should be able to achieve all these goals simultaneously. One possible direction for this is to have a more reactive "intelligent" virtual companion; however such a research direction adds numerous difficult challenges. The Virtual Companion technique could also be tested using other representations. For instance, more "human" creatures, such as a fairy or an angel, could represent a virtual "guardian angel". These creatures could justify the use of additional and promising verbal feedback. One other possible direction for an ideal interface could be the addition of other sensory feedbacks such as auditory effects (known to increase warning performances [113]) or haptic feedback (known to increase performance of rate-control [194]).

Finally, these techniques provide new metaphors that could be adapted to different VR setups. Although our user-study was performed in a CAVE-like setup, our techniques could be applied to different setups with more than one missing screen, or to Head-Mounted Displays.

7.5 Conclusion

Navigating in large VE within the confines of a VR setup with a limited translation *and* rotation workspace poses numerous additional challenges when users turn and face the missing display region. Thus, in this chapter, we presented three novel techniques that address both boundary issues with three goals in mind: keeping the user safe, providing ecological navigation, and encouraging walking.

The first metaphor, called the *Constrained Wand and Signs* extends the basic and well-known wand paradigm by adding virtual warning signs and forcing the user to walk to the limits of the space before activating the wand. The second metaphor extends the Magic Barrier Tape from Chapter 6 to deal with the problem of the missing screen and the small physical space. Our third metaphor introduces a *Virtual Companion* which uses a bird to provide an even more ecological warning and navigation metaphor.

We evaluated these three metaphors to highlight the relative advantages of each approach, and to compared them to a traditional wand "base" condition, which is a simple wand interface augmented with our novel translation virtual signs for safety reasons. We used two experimental methods to compare the techniques. First, we collected tracking data and analyzed quantities such as walking distance and speed. Second, we used Likert-scale questionnaires to evaluate the impression of users in terms of accuracy, walking sensation, and other subjective criteria. Overall, our three new techniques all achieve at least two of the initial goals. CWand achieves both safety and increased walking, eMBT achieves all three to a certain extent, while VC achieves safety and ecological interface.

We have notably introduced a comprehensive solution to the problem of seeing the missing screen in cube-like immersive spaces. This is confirmed by our study which indicates in particular that when using the eMBT metaphor users almost never look at the missing screen.

We believe that our study provides interesting insights into the various trade-offs involved in the design of navigation techniques for immersive restricted workspaces. We have introduced three new metaphors, each having different comparative strengths and weaknesses. We are particularly hopeful about the Virtual Companion paradigm and its potential for future navigation techniques.

Conclusion

In this Ph.D. manuscript, we have studied the design of multimodal feedback and interaction techniques for physically based and large VE. The main objective was to improve user interaction with complex VE within VR simulations. We followed two research axes corresponding to two fundamental interaction categories. In the first axis (Part I), corresponding to the *manipulation* category, we focused on **rendering haptic and multimodal feedback from the interaction with fluid, deformable and rigid media**. In the second axis (Part II), corresponding to the *navigation* category, we focused on **enabling immersive navigation in potentially infinite VE based on natural walking within restricted workspaces**.

To enable mutimodal feedback from different media, we first focused on the haptic and visual modalities when interacting with fluid media (Chapter 2). We proposed a **novel approach for 6DoF haptic interaction with viscous fluids**, which allows real-time 6DoF haptic interaction with fluids of variable viscosity through arbitrary shaped rigid bodies. The approach is based on the Smoothed-Particle Hydrodynamics physical model. Through a novel haptic rendering technique, we compute SPH haptic forces to produce a smooth haptic interaction. Thanks to a unified particle model, rigid bodies can interact with fluids and provide 6DoF haptic feedback. We designed different example scenarios to illustrate and evaluate some of the interaction possibilities offered by our technique. The evaluation of the technique showed the efficiency of the algorithms in VE rendered at 70Hz with up to 32,000 particles, while further implementation and hardware improvements boosted the frame rate up to 400%.

Once 6DoF rigid-fluid haptic interaction was possible, we focused on adding haptic interaction with deformable bodies to the approach, in order to obtain a **unified approach for haptic interaction with different media** (Chapter 3). Taking advantage of the flexibility of the Smoothed-Particle Hydrodynamics framework and its ability to simulate the three different states, we proposed a haptic coupling mechanism for a unified multi-state haptic rendering approach. We enabled the haptic interaction simultaneously with media in fluid, deformable and rigid states, while avoiding the complexity of dealing with different algorithms and their coupling. Thanks to an efficient dual GPU implementation, we achieve high update rates meeting the haptic requirements for each state. A perceptual experiment was conducted in order to assess the capability of users to recognize the different states of matter they interacted with. Results show a high recognition rate, even when providing only haptic feedback, and the increased appreciation of users when combining haptic and visual cues.

With our main objective being the rendering of multimodal feedback from the interaction with complex VE, we then focused on enhancing fluid interaction by generating additional feedback modalities (Chapter 4). We focused on vibrotactile and acoustic rendering, which do not require expensive devices. We proposed a **novel model for the vibrotactile rendering of fluids**, based on prior fluid sound rendering knowledge, leveraging the fact that acoustic and vibrotactile phenomena share a common physical source. The model is divided into three components: an initial impact with the fluid surface, a cavity oscillation created when the body enters the fluid, and a set of small bubble harmonics. We illustrated this approach with several fluid interaction scenarios, where users felt the fluid through vibrotactile transducers. User feedback regarding material identification solely based on vibrotactile cues suggested that the model effectively conveys the sensation of interacting with fluids, and highlighted the need for consistent kinesthetic cues.

In the first part of this manuscript we have successfully enabled the interaction with different media while rendering feedback through multiple sensory modalities. We have extensively used the powerful Smoothed-Particle Hydrodynamics framework, bringing multistate simulations to very high update rates, providing an SPH-based haptic coupling mechanism, and efficiently computing bubble-based and surface impact events. We have used different input and output devices to sense user motion and provide sensory feedback, such as 6DoF haptic devices, vibrotactile floors, hand-held vibrators, pressure sensors and audio speakers. We have designed different applications showcasing our different approaches, such as the Virtual Crepe Factory for multistate haptic interaction, and the pool and beach walking scenarios for fluid vibrotactile feedback. However, we have not addressed all sensory modalities, nor all types of media, and our approaches logically exhibit limitations inherent to emerging technologies. Nonetheless, this work opens new research directions in the interaction field, and brings seamless multimodal interaction one step closer to VR. Both future work and perspectives will be presented at the end of this chapter.

In a second part of this manuscript, we addressed the issues arising when doing natural walking in large VE within the confines of restricted workspaces. We first focused on designing a **novel metaphor for infinite walking in large VE within physical workspaces restricted in translation**, such as HMD with limited tracking range. In this context, we introduced the Magic Barrier Tape, a new navigation metaphor where the walking workspace is surrounded with virtual barrier tape in the VE (Chapter 6). We leverage the implicit “do not cross” message of barrier tape to prevent users from reaching the physical boundaries. The technique uses a hybrid position/rate control mechanism: natural walking is used inside the workspace, while rate control navigation is used to move beyond the boundaries by “pushing” on the virtual barrier tape. We conducted two experiments in order to evaluate the Magic Barrier Tape by comparing it to other state-of-the-art navigation techniques previously extended for omni-directional navigation. Results showed that the Magic Barrier Tape was faster than the other techniques, and confirmed that, by design, navigation through rate control with the Magic Barrier Tape is not meant for precise path following, but rather for coarse positioning between fine positioning tasks. Overall, the Magic Barrier Tape was more appreciated, while being more natural and less tiring.

Other VR workspaces, such as CAVE-like setups, pose additional restrictions in the form of rotational limits due to missing screens. Thus, we focused on designing **novel metaphors for infinite walking in large VE within physical workspaces restricted in translation *and* rotation**. In this new context, we presented three new techniques based on hybrid position/rate control dealing with translation and rotation boundaries (Chapter 7). These techniques are the Constrained Wand, which extends the basic and well-known wand paradigm by adding virtual warning signs, the Extended Magic Barrier Tape, which adds virtual walls to the original technique that prevent the user from looking at the missing wall, and the Virtual Companion, which uses a bird to guide the user through the VE and prevents him from reaching the boundaries. We designed and ran a user study to compare the relative advantages of each approach, and compared to a traditional wand "baseline" condition. Overall, our three new techniques all allow a safe navigation, with the Extended Magic Barrier Tape providing the most efficient solution to the missing screen problem. The study provides insights into the various trade-offs involved in navigation techniques for immersive restricted workspaces, according to different criteria such as real walking distance, naturalness, and simplicity of use.

In the second part of this manuscript, we have successfully proposed different navigation techniques based on natural walking that take into account the translational and rotational limits of the physical workspace. We designed efficient, simple and intuitive metaphors that freed the user from the use of control devices. We have used different VR setups: an HMD device with a 6 meters wide and 360° tracking system, an a CAVE setup with 4 screens and full tracking. The techniques were tested and evaluated using different VE, notably including a very rich, complex and immersive "hangar" VE. Other VR setups need to be tested, and the different approaches have room for improvement. We believe our study raised several issues in restricted navigation, and opened the area to new research directions. The remainder of this chapter addresses future work and research perspectives.

This Ph.D. thesis was conducted in the context of the FP7 European project NIW - Natural Interactive Walking. This project focused on augmenting walking in VR in terms of ground perception and navigation capabilities. It targeted novel hardware and software approaches, as well as integrated applications, with a strong emphasis on the multimodal and cross-modal perception of walking. The project involved different academic partners (Universita degli Studi di Verona, Italy, Aalborg Universitet, Denmark, Universite Paris 6, France, McGill University, Canada, and INRIA Rennes, France), leading to fruitful research collaborations.

Future Work

The work presented in this manuscript leaves some questions unanswered, which could be addressed in short-term future work. We present future research possibilities according to our three objectives presented in the introduction of this manuscript.

Concerning haptic interaction with fluid, deformable and solid media:

- **Method improvements**. Our haptic interaction approach is based on SPH unified pressure/viscosity forces computed on the proxy. These forces have the advantage of being fast to compute and unified, but do not prevent interpenetration, suffer from issues common to penalty forces and provide limited control through non physical values for the different parameters (mainly for the viscosity parameter). Other solid-fluid coupling mechanisms could be explored, leading to improved solid-fluid interaction forces. Constraint-based approaches are very attractive [195], since they avoid interpenetration while providing more control on velocities at boundaries for stick and slip conditions. In addition, the use of a virtual coupling mechanism improves the stability of the haptic loop, but introduces perceivable artificial viscosities in the feedback. Taking into account the viscosity of the virtual coupling when computing force feedback would lead to a more transparent haptic coupling.
- **Evaluation**. Further evaluations could be conducted, in order to assess the haptic perception of state changes. When evaluating our multistate approach, we used fixed stiffness and viscosity values. It would be interesting to perform state recognition

and within-state discrimination with variable stiffness for deformable bodies and variable viscosity for fluids. This could point out some limitations of our approach, and would allow application designers to take into account the perception limits when choosing the physical values of the VE.

- **Further studies**. The positive results obtained with our haptic fluid approach encourage the study of non-Newtonian fluids such as paint, starch or blood, where the viscosity coefficient is not constant, making computations much more challenging. Within deformable bodies, other interesting phenomena could be studied and rendered to the user, such as plastic deformation and tearing.

Concerning vibrotactile and acoustic feedback from fluids:

- **Method improvements**. Vibrotactile feedback coming from surrounding bubbles could be made more spatialized. Using appropriate devices with actuators distributed over the surface, each bubble oscillation generation could be routed to the appropriate actuators. This could greatly improve the interaction realism with the user and his perception of the surrounding fluid. In the case of the acoustic modality, fluid sound synthesis requires many more particles to generate a more realistic sound. Improving the simulation performance, both through a further optimized implementation and a wider use of available hardware, would allow the simulation of an increased number of particles.

- **Evaluation**. The vibrotactile interaction with fluids requires further evaluations, in order to assess different fluid and VE parameters, such as different fluid heights and different viscosities. A study on human perception thresholds also needs to be conducted for each transducer, since vibrotactile restitution and perception could greatly vary according to the device and the subject. Above all, the approach needs to be tested using vibrating shoes instead of tiles, in order to avoid the contradictory kinesthetic cues sent by the real foot-floor rigid contact instead of a virtual foot-fluid compliant contact.

- **Further studies**. The complexity and specificity of vibrotactile fluid interaction prevented us from broadening the approach to other media, such as deformable bodies. It would be interesting to interact with other media through the same modality, as well as combining different media in the simulation, such as in the beach example evoked in the introduction of this manuscript. Previous work has already proposed interesting approaches for the simulation of solid [59], granular [31], and compliant [104] materials in general. In addition, new modalities could be studied, using cross-modal information such as when generating pseudo-haptic feedback [196].

Concerning navigation metaphors for walking in restricted workspaces:

- **Method improvements**. Our different navigation metaphors could be improved by throughly tuning speed, threshold and gestures parameters based on user feedback from our studies. This alone could already improve the techniques efficiency and easiness of use. In addition, the metaphors could be extended with mechanisms that force the user to walk more in the real workspace. Possible approaches range from the addition of inertial or automatic translation and rotation movements to the combination with redirection techniques.

- **Evaluation**. Our approaches were evaluated using generic VE with generic pointing and path following task. For real use cases, it would be interesting to conduct evaluations in specific VE with specific contextual tasks (maintenance, training) which require walking. It might be possible that not all metaphors are adequate for a given task. Other studies are required to assess the performance and usability of the techniques in more constrained VR setups, such as workbenches and 3-face CAVEs.

- **Further studies**. For a much broader use, these techniques could be adapted to the most basic VR setups (single frontal screen with stereo, such as geowalls). The challenge lies in the lack of floor projection: stereoscopic perception is lost for objects standing out of the screen, since they will be cropped by the single screen display. Since techniques rely on the use of an immersive setup, the task is far from trivial. In addition, the Virtual Companion concept could be further developed by leveraging the presence of a companion for tasks other than protection. The companion, which could be a more “human-like” creature, could provide guidance in the VE (such as where to go), convey additional and complementary information, and, in general, generate more contextual interaction.

Long-Term Perspectives

In addition to the future work mentioned above, this Ph.D. thesis also paved the way for new research directions and long-term views. Some of these aspects are described below.

Towards fully multimodal VE with low-level integration

There is a growing interest in the entertainment consumer market in improving user immersion. The usual visual and acoustic modalities have seen enhancements that are becoming widely available, while evolving into industry standards: stereoscopic vision displays (TV screens, video projectors, portable gaming consoles) can be bought off-the-shelf and are found in many homes, while affordable surround sound systems have been around for many years now. Many of nowadays top selling video games exploit the capabilities of these enhanced devices to improve the overall experience by increasing user immersion.

However, it is interesting to notice that modalities other than the aforementioned are starting to gain the attention of industrials and consumers. All major gaming systems include vibrotactile transducers inside their controllers. Novint’s Falcon 3DoF device shows that force-feedback devices, which have always been expensive interfaces, are now being commercialized in the mass market at affordable prices. Microsoft Kinect, a tracking interface designed to allow users to interact with the system through their bodies, enhance vestibular and proprioceptive cues during interaction. All these devices contribute to a multimodal and immersive experience.

Their rendering capabilities, however, are still limited. This is in part due to hardware limitations, but also from the unavailability of multimodal models allowing the simulation of phenomena and their rendering through many different sensory channels. Thus, we believe the multimodality aspect should be an important component of the design of novel VR simulations. Future research could focus on developing models with a tight low-level integration of rendering modalities. This would free developers from implementing one specific model for each rendered modality, greatly reducing on-line computational demands, development times and costs, while improving the overall experience. Our fluid multimodal interaction approach is an example of such a low-level integration, with only one simulated model allowing kinesthetic, vibrotactile and acoustic feedback. Users could interact with water while perceiving the fluid resistance, its bubbly feeling and characteristic sound.

Future interaction devices could also be designed with multimodality in mind once these integrated models start to emerge. Foot-based devices, such as Nintendo's Wii Balance Board or novel actuated shoes, could be equipped with vibrotactile transducers, allowing users to step into different grounds, such as water puddles and sandy or snowy VE. Hand-held controllers could be made of different moving mechanical parts, or of variable stiffness materials, allowing to generate kinesthetic compliance effects for the interaction with deformable objects. In addition, not only the entertainment industry could benefit from multimodal interaction. All applied areas of VR could be targeted, such as in training simulations and virtual walkthroughs for museums and architectural projects, to name a few.

In the longer term, and with the advent of olfactory [108] and gustatory [109] devices, new models could explore the integration of these two last senses. However, it is not clear at which level of the model smell and taste rendering algorithms could be integrated, since these phenomena come from a different physical source. However, the ever increasing computational power might allow us to simulate matter at microscopic scales, allowing us to design fully multimodal unified models with an integration at the molecular level.

Towards navigation of large VE in arbitrary-shaped and dynamic workspaces

Although the navigation metaphors that were presented in this manuscript could be adapted to other setups, we designed and evaluated them with well-defined workspaces in mind. These include smoothly shaped tracking areas and cube-like setups. For current uses of VR systems in research laboratories and industry, this context is probably sufficient.

However, VR is getting outside of labs and industries. There is an increased availability of low-cost devices and home-made setups, and, as previously mentioned, a commercial growth of VR and interaction thanks to gaming and simulation in general. At home, users can have HMD setups for reasonable prices, and low-cost tracking systems are becoming widespread, with the latest example being the Microsoft Kinect. VR technology can also be setup in public places, for exhibitions and demonstrations, or in locations where space is an issue, such as offices or oddly shaped rooms. A major problem that arises is that, in these contexts, workspaces are not as convenient as in specifically designed places. Workspaces might be ill-shaped (closed angles, narrow areas), might not be fully navigable (obstacles in the way, sensitive and dangerous areas), and might not be static (people could be passing by in public areas, obstacles and boundaries could be moved around). Manifestly, these workspaces exhibit a rather unfriendly context for natural walking navigation. However, the navigation metaphor goals remain the same: allow safe navigation using natural walking without breaks of immersion.

Thus, future research in VR navigation techniques could focus on providing safe and immersive metaphors for restricted, arbitrary-shaped, non-empty and dynamic workspaces. This challenging work would probably require the creation of a dynamic map of the real environment, the integration of artificial intelligence algorithms for predicting behaviors and potential walking paths, context-dependent knowledge for the anticipations of events, collision detection and avoidance algorithms, as well as some approaches borrowed from the field of collaborative VR. All this should be integrated in a common, simple, intuitive, safe and immersive navigation metaphor, leveraging natural cues as with the Virtual

Companion presented in this manuscript.

Towards the integration of manipulation and navigation techniques in complex VE

The ever-increasing computational power and research advances will lead us towards increasingly complex VE, and increasingly higher user capabilities. Users will eventually want to replicate their real life actions and situations without compromise. In this manuscript, we have addressed manipulation and navigation issues in complex VE in two distinct tracks. On one side, we have novel haptic and multimodal rendering techniques for the manipulation of physically based VE. On the other, we have navigation metaphors for large VE. However, the combination of both, namely manipulating a virtual object with multimodal feedback during a navigation task, is something that is seamlessly and commonly done in real life, but that still needs to be addressed in VR.

First of all, rendering devices that have user proximity constraints, such as haptic, olfactory and gustatory devices, needs to follow the user during navigation. *Manipulation* needs to be enhanced with *navigation.* Taking user motion into consideration, future research could design more devices that can be worn by the user (such as a vibrotactile glove), or laid out on the entire workspace (such as an actuated floor). Others devices, such as grounded kinesthetic devices, would require motion platforms to follow the user. Although many contributions have already been done in this area, there are still many engineering and scientific problems that remain to be solved.

Second, navigation could take into account floors and terrains as any other *manipulated* object of the VE. *Navigation* needs to be enhanced with *manipulation.* This would allow the generation of feedback during navigation tasks. In addition, the different physically based ground behaviors could influence user navigation: locomotion could get naturally harder due to compliant, frictional, viscous or sticky grounds, virtual obstacles in the path could stop user navigation with physically based feedback, terrain slopes could be taken into account with an effect on locomotion through a physically based biomechanical model. The integration of navigation and manipulation techniques would allow the simulation and restitution of many real world behaviors, thus significantly improving user immersion in high-end VR applications.

Naturally, a lot of work is still required in order to achieve seamless multimodal interaction with any kind of VE. As VR interaction benefits from technology improvements in devices and computational power, the next decades will witness tremendous advances in the field and, at the same time, will open the way to new and exciting research challenges. We hope that the work presented in this manuscript will contribute in this context, building the bridge between real and virtual worlds.

A

Appendix: Fundamentals of Physically Based Simulation Models for Haptic Interaction

Contents

A.1 Rigid bodies **151**

A.2 Deformable bodies **153**

  A.2.1 Continuum mechanics 153

  A.2.2 The Finite Element Method 154

  A.2.3 Mass-spring systems 156

A.3 Fluids **157**

  A.3.1 Navier-Stokes equations 157

  A.3.2 Eulerian simulation 158

A.4 Time integration schemes **161**

  A.4.1 Explicit integration 162

  A.4.2 The implicit Euler method 163

In this appendix, we present the fundamentals of the different physically based models used in haptic rendering techniques for the simulation of rigid bodies, deformable bodies and fluids. Rigid bodies follow rigid body dynamics, described in Section A.1. Deformable bodies follow the laws of continuum mechanics, described in Section A.2, through two main discretization models: the Finite Element Method (Section A.2.2) and mass-spring systems (Section A.2.3). Fluids follow the Navier-Stokes equations, described in Section A.3, and are discretized using the Eulerian approach described in Section A.3.2. This appendix also covers in Section A.4 the different explicit and implicit integration schemes used in previous haptic work. For further details on the different models and schemes, we refer the reader to [106].

A.1 Rigid bodies

A good and comprehensive introduction to rigid body simulation is given by Baraff in [197]. Rigid bodies can be modeled as an infinite set of particles with position $\mathbf{x}_i$, velocity $\mathbf{v}_i$ and mass m_i. A rigidity assumption forces a constant relative distance between particles, as if they were linked by springs of infinite stiffness. This assumption greatly simplifies the dynamics of the rigid body, whose state can be characterized by only a few dynamic variables applied at the center of mass $\mathbf{x}$ of the body: the position $\mathbf{x}$ (corresponding to the center of mass itself), the linear velocity $\mathbf{v}$, the orientation R and the angular velocity ω. The position of the center of mass is computed as:

$$\mathbf{x} = \frac{1}{m}\sum_i m_i \mathbf{x}_i \tag{A.1}$$

where m is the total mass of the body ($m = \sum_i m_i$).

Another important rigid body component is its inertia matrix J. The inertia matrix is to angular motion what the mass is to linear motion. J depends on the rigid body orientation, but can computed at each time step using J_{init}, the inertia matrix generated with the initial particle positions, and the current orientation R, following $J = RJ_{init}R^T$. J_{init} is computed for $R = I_3$ as:

$$J_{init} = \sum_i -m_i skew(\mathbf{r}_i) skew(\mathbf{r}_i) \tag{A.2}$$

where $\mathbf{r}_i = \mathbf{x}_i - \mathbf{x}$, and $skew(\mathbf{r}) \in \mathbb{R}^{3\times3}$ is the matrix with the property $skew(\mathbf{r})\mathbf{x} = \mathbf{r} \times \mathbf{x}$.

Newton's second law of motion provides a relationship between the total force $\mathbf{f}$ applied on the rigid body and its linear acceleration:

$$\mathbf{f} = m\dot{\mathbf{v}} \tag{A.3}$$

This is also the case for angular motion, with a relationship between the total torque τ applied on the rigid body and its angular acceleration:

$$\tau = J\dot{\omega} \tag{A.4}$$

Both $\mathbf{f}$ and τ are computed by summing the contributions of each particle of the rigid body, $\mathbf{f} = \sum_i \mathbf{f}_i$ and $\tau = \sum_i \mathbf{f}_i \times \mathbf{r}_i$, where $\mathbf{f}_i$ is the force applied at particle i.

Both position and orientation can be regrouped in a state vector $S \in \mathbb{R}^{12}$:

$$S = \begin{bmatrix} \mathbf{x} \\ R \end{bmatrix} \tag{A.5}$$

while velocities are given by $\dot{S} \in \mathbb{R}^{12}$:

$$\dot{S} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ skew(\omega)R \end{bmatrix} \tag{A.6}$$

The total force and torque can be regrouped into a generalized force vector $\mathbf{f}_{ext} \in \mathbb{R}^6$, while the mass and the inertia matrix can be regrouped into a mass matrix $M \in \mathbb{R}^{6\times6}$ as follows:

$$\mathbf{f}_{ext} = \begin{bmatrix} \mathbf{f} \\ \tau \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} m^{-1} & & & 0 \\ & m^{-1} & & \\ & & m^{-1} & \\ 0 & & & (RJ_{init}^{-1}R^T) \end{bmatrix} \tag{A.7}$$

With these notations, equations A.3 and A.4 can be written as the following partial differential equation:

$$\ddot{S} = M^{-1}\mathbf{f}_{ext} \tag{A.8}$$

$\ddot{S}$ can be solved and integrated over time using different time integration techniques (see Section A.4) to obtain a new rigid body state.

A.2 Deformable bodies

A.2.1 Continuum mechanics

Deformation was introduced in the computer graphics community more than twenty years ago, thanks to the pioneering work of Terzopoulos and colleagues [198], proposing methods to simulate elastic and plastic deformation and fracture using continuum mechanics. A comprehensive review on deformation modeling and simulation in computer graphics can be found in [199].

In continuum mechanics, a three dimensional deformable object is thought as a continuous connected subset $\Omega \in \mathbb{R}^3$. The object is typically defined by its undeformed (rest or initial) shape and by a set of material parameters that define how it deforms under applied forces. The coordinates $\mathbf{x} \in \Omega$ of a point in the undeformed shape of the object are called *material coordinates*. When forces are applied, the object deforms and a point originally at location $\mathbf{x}$ moves to a new location $\mathbf{p}(\mathbf{x})$, called the spatial or world coordinates. The deformation is represented by the displacement field $\mathbf{u}(\mathbf{x}) = \mathbf{p}(\mathbf{x}) - \mathbf{x}$.

Besides displacement there are two other main quantities to model elastic deformation: the stress σ, measuring the average force per unit area of a surface within the body on which internal forces act, and the strain ϵ, a description of deformation in terms of relative displacement (elongation or compression) of the material. The relation between stress and strain, called constitutive law, can be approximated using Hooke's Law:

$$\sigma = E\epsilon \tag{A.9}$$

which states that stress and strain are linearly related. Many materials follow this law, thus called "Hookean materials", provided deformations remain small. For isotropic materials (with equal behavior in all directions), and in the three dimensional case, $E \in \mathbb{R}^{6\times 6}$ is equal to:

$$E = \frac{\lambda}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & & & 0 \\ \nu & 1-\nu & \nu & & & \\ \nu & \nu & 1-\nu & & & \\ & & & 1-2\nu & & \\ & & & & 1-2\nu & \\ 0 & & & & & 1-2\nu \end{bmatrix} \tag{A.10}$$

where λ is the elastic stiffness of the material (Young's modulus) and ν the amount of volume conservation within the material (Poisson's ratio).

The strain $\epsilon \in \mathbb{R}^{3\times 3}$ is derived from the spatial derivatives of the displacement field $\mathbf{u}(\mathbf{x})$. Two common approaches to computing the strain are Green's (non-linear, ϵ_G) and Cauchy's (linear, ϵ_C) strain tensors:

$$\epsilon_G = \nabla\mathbf{u} + [\nabla\mathbf{u}]^T + [\nabla\mathbf{u}]^T\nabla\mathbf{u}) \tag{A.11}$$

$$\epsilon_C = \nabla\mathbf{u} + [\nabla\mathbf{u}]^T \tag{A.12}$$

where $\nabla \mathbf{u} \in \mathbb{R}^{3\times 3}$ is the gradient of the displacement field.

From the stress tensor $\sigma \in \mathbb{R}^{3\times 3}$, one can compute a force $\mathbf{f} \in \mathbb{R}^3$ acting on an area a in a direction $\mathbf{n}$, i.e. the force per area acting on a plane whose normal vector is $\mathbf{n}$:

$$\frac{\mathbf{f}}{a} = \sigma \cdot \mathbf{n} \tag{A.13}$$

By extension, if we take the forces acting on the faces of an infinitesimal element, we arrive at the final expression for the body forces $\mathbf{f}_\sigma \in \mathbb{R}^3$ acting on the element due to internal stresses:

$$\mathbf{f}_\sigma = \nabla \cdot \sigma \tag{A.14}$$

Using Newton's law, the partial differential equation (PDE) governing dynamic elastic materials can be written down (since the mass of an infinitesimal volume is not defined, density ρ and body forces $\mathbf{f}_\sigma$ are used instead):

$$\rho \ddot{\mathbf{p}} = \mathbf{f}_\sigma + \mathbf{f}_{ext} \tag{A.15}$$

where $\mathbf{f}_{ext}$ are externally applied body forces such as gravity.

The goal is to compute $\mathbf{p}$, the new positions of the nodes of the deformable body, at each time step. The density ρ and the external forces $\mathbf{f}_{ext}$ are known quantities. The first step is the computation of the displacement field $\mathbf{u} = \mathbf{p} - \mathbf{x}$, from which the strain field ϵ can be obtained following eqs. A.11 or A.12. Then, the stress field σ is derived from the strain using eq. A.9. After using σ to compute the stress forces $\mathbf{f}_\sigma$ using eq. A.14, the PDE from eq. A.15 allows the computation of the acceleration $\ddot{\mathbf{p}}$ of the nodes, which in turn define how the positions evolve.

Although it is mathematically possible to model the deformation as eq. A.15, it is numerically impossible to solve since we can not manage infinite sets. In addition, so far, this equation has no analytical solution. That is why the Finite Element Method (FEM) is derived from continuum mechanics, spatially and temporally discretizing the material properties.

A.2.2 The Finite Element Method

In order to solve eq. A.15, the Finite Element Method (FEM) proposes to partition the continuum into a set of non-overlapping elements. Each element has a finite set of nodes, with each node carrying the three quantities previously presented (the displacement, the strain and the stress) that enable to simulate deformation. Inside the element, we use a linear mapping to interpolate the needed quantities.

Let $\mathbf{x} \in \mathbb{R}^3$ be a point inside a tetrahedral element of the deformable body, itself defined by four nodes of position $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ and $\mathbf{x}_4$ in the undeformed state. We can formulate the vector $\mathbf{x}$ as a linear combination of $\mathbf{x}_1..\mathbf{x}_4$ using four barycentric coordinates:

$$\mathbf{x} = \mathbf{x}_1 b_1 + \mathbf{x}_2 b_2 + \mathbf{x}_3 b_3 + \mathbf{x}_4 b_4 = [\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3\ \mathbf{x}_4]\mathbf{b} = A\mathbf{b} \tag{A.16}$$

In the deformed state, the same nodes have different coordinates, which are $\mathbf{p}_1..\mathbf{p}_4$ respectively. The position $\mathbf{p}$ of $\mathbf{x}$ in the deformed state is computed by interpolation using the same barycentric coordinate as in the undeformed state:

$$\mathbf{p} = \mathbf{p}_1 b_1 + \mathbf{p}_2 b_2 + \mathbf{p}_3 b_3 + \mathbf{p}_4 b_4 = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4]\mathbf{b} = B\mathbf{b} \tag{A.17}$$

Thus, we have $b = A^{-1}\mathbf{x}$ and $b = B^{-1}\mathbf{p}$, yielding $B^{-1}\mathbf{p} = A^{-1}\mathbf{x}$, or:

$$\mathbf{p} = BA^{-1}\mathbf{x} = Px \tag{A.18}$$

A is constant over time since it is built from the undeformed position of the nodes. Since $P \in \mathbb{R}^{3\times3}$ is a linear mapping for position, the gradient of the displacement field is:

$$\nabla\mathbf{u} = P - I \tag{A.19}$$

where $I \in \mathbb{R}^{3\times3}$ is the identity matrix. Thus, the gradient of the displacement field is constant within an element, yielding constant strain and stress fields per element.

Using eq. A.11 or A.12 we can compute the stress ϵ, and the strain σ using eq. A.9. Multiplying the stress tensor by a normal vector $\mathbf{n}$ yields the elastic force per area. For the face with nodes (1, 2, 3) of the tetrahedron the force is:

$$\mathbf{f}_{1,2,3} = \sigma \cdot \mathbf{n}_{1,2,3} \cdot A_{1,2,3} = \sigma[(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)] \tag{A.20}$$

Finally, we distribute this force evenly among the nodes of the face, and do the same for all faces. Thus, eqs. A.18 through A.20 allow the computation of forces acting on the nodes of a tetrahedron based on the deformed positions $\mathbf{p}_1..\mathbf{p}_4$. Using an explicit integration algorithm and eq. A.15, the new positions p can be computed.

However, in order to use implicit integration for unconditional stability, a different approach has to be adopted. A tangent stiffness matrix $K_e \in \mathbb{R}^{12\times12}$ has to be computed for each tetrahedron. This matrix contains 4×4 submatrices $K_{i,j}$, where $K_{i,j}$ describes the interaction between node i and node j, i.e. the gradient of the force exerted on node i knowing the position of node j and assuming the other nodes are at their rest position. Since each node in the tetrahedron influences all others, the tangent stiffness matrix of the tetrahedron is dense. With this matrix, the tetrahedral forces $\mathbf{f}_e$ can be approximated as

$$\mathbf{f}_e(\mathbf{p} + \Delta t\mathbf{v}) = \mathbf{f}(\mathbf{p}) + K_e\Delta t\mathbf{v} + O(\Delta t^2) \tag{A.21}$$

where all vectors are 12 dimensional containing the 3 components of the 4 adjacent vertices of the tetrahedron.

Similarly, considering all the elements of a discretized deformable body, it is possible to assemble a global stiffness matrix K from all the K_e matrices of each element (since a node can belong to several elements, the assembling of the matrices K_e will have overlapping parts, which are summed up to obtain K). Defining a global force vector $\mathbf{f}$ as the concatenation of all the forces $\mathbf{f}_e$ and a global displacement vector $\mathbf{u}$ as the concatenation of all the displacement vectors, we can write:

$$\mathbf{f} = K(\mathbf{u})\mathbf{u} \tag{A.22}$$

The stiffness matrix K can be computed for the entire deformable body at each step of the simulation from the deformed state of the body, and the new unconditionally stable values of p can be obtained through implicit time integration (see Section A.4). However, computing the stiffness matrix at each time step can be prohibitively expensive for real-time simulations. Thus, linear FEM can be used under small deformations.

A.2.2.1 Linear FEM

In linear FEM, the stiffness matrix K is evaluated at the rest shape, and remains constant throughout the simulation. Thus, it can be precomputed, greatly increasing computation times at runtime. In the rest condition, $\mathbf{f}(\mathbf{p})$ in eq. A.21 cancels out since there are no elastic forces at rest, leading to:

$$\mathbf{f}(\mathbf{p}) = K\mathbf{u} \tag{A.23}$$

This approximation is only valid close to the equilibrium, since that is where the forces were linearized. Since forces are linear, we can use Cauchy's strain tensor (eq. A.12) and assume the material is hookean (eq. A.9). For the final form of K, we refer the reader to [106]. The main restriction of this approach is that displacements have to remain small, otherwise unnatural deformations and artifacts appear. The corotated approach proposes a way of getting rid of these artifacts, allowing large deformations through linear FEM.

A.2.2.2 Corotated formulation

The corotated approach [102] works by separating the rigid transformation from the non-rigid deformation when computing the forces acting on each node. It is possible to find a 3×3 rotation matrix R_i which represents the rotational part of a deformed node i. The nodal forces are computed using the unrotated node positions $R_i^{-1}\mathbf{p}_i$ of each node. Most of the time, we choose to have the same rotation matrix R_e for each node of the same element e. This enables to write an element-based rotation using $R_e \in \mathbb{R}^{12\times 12}$, which is formed putting together 16 (4 in the width, and 4 in the height) copies of the 3×3 rotation matrix R_i. The nodal force computation now becomes:

$$\mathbf{f}_e = R_e K_e R_e^{-1}\mathbf{u}_e \tag{A.24}$$

The displacement field is first unrotated, then local forces are computed using the tangent stiffness matrix K_e. The forces are transformed back into the rotated state of the deformed element.

There are several ways to determine the rotation matrix R_e. It can be computed from the transformation matrix P of the element defined in eq. A.18 using, for instance, the Gram-Schmidt method or polar decomposition [102]. Corotational formulations enable fast and stable simulations of large deformations, without the artifacts of linear FEM.

A.2.3 Mass-spring systems

Mass-spring systems implicitly model deformation in a simple way. Their main idea is to discretize the matter into a finite set of particles. Each particle i has a mass m_i and a position $\mathbf{x}_i \in \mathbb{R}^3$. The particles are connected with springs. Each spring s has a stiffness k_s, a damping factor d_s, a rest length l_{0_s}, and two more scalars representing the index of the two particles that are linked through the spring.

The particles are submitted to the classical dynamic equilibrium $\ddot{\mathbf{x}}_i = \frac{\mathbf{f}_i}{m_i}$, where $\mathbf{f}_i \in \mathbb{R}^3$ is the total force acting on particle i. We can decompose $\mathbf{f}_i$ into $\mathbf{f}_i = \mathbf{f}_{ext_i} + \mathbf{f}_{spr_i}$, where $\mathbf{f}_{ext_i}$ is the sum of external forces such as gravity or collision forces, and $\mathbf{f}_{spr_i}$ is the sum of internal forces of the body coming from the spring system. Both forces act on particle i. If $\jmath \subset \mathbb{R}$ is the set of particles which are neighbors of particle i (i.e. which have a spring linking to particle i), we can write:

$$\mathbf{f}_{spr_i} = \sum_{j \in \jmath} \left[\left(k_s(|\mathbf{x}_j - \mathbf{x}_i| - l_{0_s}) \frac{(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|} \right) - \left(d_s(\ddot{\mathbf{x}}_j - \ddot{\mathbf{x}}_i) \cdot \frac{(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|} \right) \right] \tag{A.25}$$

In eq. A.25, the first right term represents the force acting along the direction of the spring and proportional to the difference of length between the current and the rest lengths of the spring. The second right term is the damping factor, proportional to the relative velocity of the particles along the spring.

Mass-spring systems are widely used due to their simplicity and their cheap computation requirements. In addition, they are easy to implement. However, their main drawback comes from the fact that the behavior of the simulated body depends on the spring configuration, and their individual stiffness and damping factors. In addition, volumetric constraints such as volume conservation cannot be directly handled by mass-spring systems. Those drawbacks make mass-spring systems very hard to configure in order to achieve the desired behavior.

A.3 Fluids

A.3.1 Navier-Stokes equations

A detailed introduction to fluid simulation for computer graphics can be found in Bridson's course [126] and book [200].

The general incompressible Navier-Stokes equations dictate the motion of an incompressible Newtonian fluid:

$$\frac{D\mathbf{v}}{Dt} + \frac{1}{\rho}\nabla p = \mathbf{f} + \nu \nabla \cdot \nabla \mathbf{v}, \tag{A.26}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{A.27}$$

where $\mathbf{v}$ is the fluid velocity field and $\frac{D\mathbf{v}}{Dt}$ its *material* derivative, ρ is the density of the fluid ($\rho \approx 1000$ kg$/m^3$ for water), p the pressure field, $\mathbf{f}$ the body forces (such as gravity) per unit mass and ν the coefficient of kinematic viscosity of the fluid.

Equation A.26 is the momentum equation, which tells how the fluid accelerates due to body, pressure and viscosity forces acting on it. Equation A.27 is the incompressibility condition, ensuring the fluid volume does not change.

Fluid viscosity is often not taken into account, as in some cases of previous work on haptic interaction with fluids. In many cases, the numerical damping introduced by numerical resolution errors already give a viscous behavior to the fluid motion. The resulting equations for inviscid motion, after dropping the viscosity term, are called the Euler equations, and are simpler to solve.

There are two viewpoints for the simulation of fluids: the Lagrangian, and the Eulerian. The Lagrangian viewpoint treats fluid as a particle system, with each particle holding a set of fluid quantities such as position, velocity, mass and viscosity. The particles (and hence the associated values) move in space, creating the overall fluid motion. The benefits of the Lagrangian viewpoint is the simple enforcement of mass conservation (since held as an explicit quantity by particles), and faster computations. The Eulerian viewpoint, on the other hand, tracks the fluid quantities at specific and fixed points in space as fluid flows by. The Eulerian viewpoint uses grid based techniques to manipulate the different quantities at fixed points. It has the advantage of having a higher numerical accuracy, since fixed grids are more appropriate than particles to work with spatial derivatives, and can easily deal with volume conservation. However, it suffers from mass loss, and is slower to compute than particle based simulations.

In the haptics literature, only the Eulerian viewpoint has been used in fluid simulation and force feedback computation. Thus, we detail this approach in the following section.

A.3.2 Eulerian simulation

From an Eulerian viewpoint, eq. A.26 becomes:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + \frac{1}{\rho} \nabla p = \mathbf{f} + \nu \nabla \cdot \nabla \mathbf{v} \tag{A.28}$$

due to the expansion of the material derivative. The new term that appeared on the left side of eq. A.28 is the advection term, measuring the rate of change of a quantity (in this case $\mathbf{v}$) at a fixed point in an Eulerian context.

The type of grid used to sample and store the different fluid quantities plays an important role. The marker-and-cell (MAC) [201] staggered grid is usually employed: a grid where pressure quantities are stored at the center of the cells, but velocity quantities are stored at the center of the faces of the cells, split into their three dimensional Cartesian components. This data structure allows the efficient use of central differences to sample the spatial derivatives of some quantities playing a major role in the Navier-Stokes equations.

These differential equations are solved by splitting them into separate components and solving each one of them separately. Thus, in the context of computer graphics and previous haptic work, they are split into 3 or 4 components (depending on whether viscosity is taken into account) and thus solved in an equal number of subsequent steps, where the velocity output of each step is used as input of the next step. For now, we focus on inviscid fluids, and thus have 3 steps, namely the addition of body forces, advection, and projection:

$$\mathbf{v}_0 \xrightarrow{addforce} \mathbf{v}_1 \xrightarrow{advect} \mathbf{v}_2 \xrightarrow{project} \mathbf{v}_3$$

Adding body forces

Body forces are forces that are applied throughout the entire volume of fluid, not just on the surface. The only body force used here is gravity $\mathbf{g}$. Thus, adding body forces as a first step boils down to solving:

$$\frac{\partial \mathbf{v}_1}{\partial t} = \mathbf{g} \tag{A.29}$$

For this, a simple forward Euler explicit integration is used (see Section A.4):

$$\mathbf{v}_1 = \mathbf{v}_0 + \Delta t \mathbf{g} \tag{A.30}$$

Advection

The second step to solve is advection, represented by the following equation:

$$\frac{D\mathbf{v}_2}{Dt} = 0 \tag{A.31}$$

During this step, we need to know how the velocity field $\mathbf{v}$ changed at each point in the grid when moved through the fluid velocity field itself. Any other quantity transported by the grid (such as smoke particles and temperature) should be advected as well.

A classic and stable approach is to use semi-Lagrangian advection involving backward particle trace. The idea consists in sampling the quantity of the fictitious particle P of the previous time step that ended up in the grid point G we are looking at in the current time step. The approach is semi-Lagrangian, since we are using particles, but these particles are fictitious. Thus, in order to find the position $\mathbf{x}_P$ from where to sample the velocity, we can use forward Euler:

$$\mathbf{x}_P = \mathbf{x}_G - \Delta t \mathbf{v}_1(\mathbf{x}_G) \tag{A.32}$$

Other techniques, such as the Runge-Kutta method (see Section A.4), can give a more accurate result. Since $\mathbf{x}_P$ is probably not a grid point, the quantities from neighboring grid points are interpolated to find the resulting advected velocity $\mathbf{v}_2$. Thus, the velocity field at each grid point can be advected, as well as any other transported quantity.

Projection

The projection step combines the last component, involving pressure, with the incompressibility constraint:

$$\frac{\partial \mathbf{v}_3}{\partial t} + \frac{1}{\rho} \nabla p = 0 \tag{A.33}$$

such that

$$\nabla \cdot \mathbf{v}_3 = 0$$

Here, the new pressure values are computed while enforcing incompressibility, and the new velocity field can be computed from the new pressure values. It is called projection, since it “projects” the velocity field onto a divergent-free field.

To this end, and following eq. A.33, the new velocity values $\mathbf{v}_3$ are first formulated as a function of current velocity values $\mathbf{v}_2$ and pressure values p, using central difference approximations for each sampled velocity. In 2D, this leads to:

$$u_3^{i+1/2,j} = u_2^{i+1/2,j} - \Delta t \frac{1}{\rho} \frac{p^{i+1,j} - p^{i,j}}{\Delta x}, \tag{A.34}$$

$$u_3^{i,j+1/2} = u_2^{i,j+1/2} - \Delta t \frac{1}{\rho} \frac{p^{i,j+1} - p^{i,j}}{\Delta x} \tag{A.35}$$

The superscripts i and j for p designate the grid cell coordinates, with p sampled at their center. Since the grid is staggered, half a step coordinates for u such as $i + 1/2$ designate the side face along i of grid cell i, j where velocity is sampled. Δx is the grid cell size.

The incompressibility constraint, which is a divergence operator applied on the velocity field, is the sum of the partial spatial derivatives of u, and thus can be formulated in 2D as:

$$(\nabla \cdot \mathbf{v}_3)^{i,j} \approx \frac{u_2^{i+1/2,j} - u_2^{i-1/2,j}}{\Delta x} + \frac{u_2^{i,j+1/2} - u_2^{i,j-1/2}}{\Delta x} \tag{A.36}$$

By plugging eqs. A.34 and A.35 into eq. A.36, we obtain a linear system with unknowns p. This system can be put in a matrix vector form of type $Ap = b$, and solved through different numerical methods specific to this problem. In previous fluid haptic work, an iterative solver called the modified incomplete Choleski Conjugate Gradient solver is used, exploiting the many specific properties of the system to be solved (symmetric positive definite, among others).

A.3.2.1 Extensions

Viscous fluids

The viscosity term was neglected in the first part of this section, since numerical errors already introduce damping in the fluid motion, resulting in a viscous behavior. However, this term is required for highly viscous fluids. In the Eulerian viewpoint, the viscous term can be solved as an additional step, diffusion, between advection and projection:

$$\mathbf{v}_0 \xrightarrow{addforce} \mathbf{v}_1 \xrightarrow{advect} \mathbf{v}_2 \xrightarrow{diffuse} \mathbf{v}_3 \xrightarrow{project} \mathbf{v}_4$$

with the following equation to solve:

$$\frac{\partial \mathbf{v}_2}{\partial t} = \nu \nabla \cdot \nabla \mathbf{v}_2 \tag{A.37}$$

This step is called diffusion since eq. A.37 is equivalent to a diffusion equation. This is a standard type of equation, and many numerical procedures can solve it. In order to have an unconditionally stable resolution, an implicit method can be used, approximating it with:

$$(I - \nu \Delta t \nabla \cdot \nabla)\mathbf{v}_3 = \mathbf{v}_2 \tag{A.38}$$

where I is the identity operator.

When discretized using the grid, eq. A.38 leads to a sparse linear system with unknowns $\mathbf{v}_3$, which can be solved using iterative approaches as in the projection step.

Rigid-fuid coupling

There are several rigid-fluid coupling schemes in the Eulerian simulation of fluids. In the haptic literature, the simplest approach is used: the rigid object is voxelized into the grid, normal boundary fluid velocities are directly updated from rigid velocities, and forces are computed on each rigid cell based on the resulting pressure and velocity values.

In order to incorporate the (Lagrangian) rigid body into the Eulerian fluid grid, the rigid body has to be voxelized using the same cells as the grid. Thus, grid cells can be flagged as entirely fluid or rigid. Then, a boundary condition has to be enforced at the cell faces shared by fluid and rigid cells. This condition states that fluid velocities $\mathbf{v}$ and rigid body velocities $\mathbf{v}_r$ are equal along the face's normal $\mathbf{n}$, which ensures non-penetration:

$$\mathbf{v} \cdot \mathbf{n} = \mathbf{v}_r \cdot \mathbf{n} \tag{A.39}$$

Since velocities are defined at the cell's faces in the staggered grid, this condition can be satisfied by directly updating the fluid velocity at the boundary faces with the rigid body velocity projected onto each face's normal.

Then, for a two-way coupling, the total force and the total torque exerted by the fluid on the rigid body are computed. The force per unit area **P** at a point **x** in the fluid on an infinitesimal area with normal **n** is given by:

$$\mathbf{P} = \sigma(\mathbf{x}) \cdot \mathbf{n} \tag{A.40}$$

where $\sigma(\mathbf{x})$ is the fluid stress tensor defined by pressure and viscosity components, leading to the following formulation of eq. A.40 in 2D Cartesian coordinates:

$$\mathbf{P} = \begin{bmatrix} -pn_x + \rho\nu(2\frac{\partial u_x}{\partial x}n_x + (\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x})n_y) \\ -pn_y + \rho\nu(2\frac{\partial u_y}{\partial y}n_y + (\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x})n_x) \end{bmatrix} \tag{A.41}$$

where the x and y subscripts denote vector components along each dimension.

Eq. A.41 is then discretized using the grid, giving force values for each boundary cell along the boundary normal. Each boundary cell force contribution is summed to obtain the total force exerted by the fluid on the rigid body. Similarly, the total torque can be computed by summing the torque contribution of each boundary cell. Torque contributions are given by the cross product between **P** and the distance vector to the rigid body center of mass or, in the case of a haptic proxy, to the rigid body attach point.

A.4 Time integration schemes

Time integration schemes compute the evolution of the state of the VE with respect to time. An overview of time integration schemes commonly used in computer graphics can be found in [106].

The global position and velocity state and the forces acting on the VE are linked in time, having the following time-differential relation:

$$M\dot{\mathbf{v}} = \mathbf{f}(\mathbf{x}, \mathbf{v}) \tag{A.42}$$

which is Newton's second law of motion put into a generalized form. The matrix M is the global inertia matrix of the system. Thus, eq. A.42 is a second order differential equation that governs the motion of the VE. The unknown is the acceleration vector $\dot{\mathbf{v}}$, while the total forces **f** and the inertia matrix M are known.

The VE is simulated forward in time by intervals of time Δt called time steps. The time step depends on different parameters and conditions, such as update rate (a higher update rate will use a smaller time step) and stability (in some integration schemes, the time step must not exceed a certain value in order to keep the system stable), and is not necessarily constant throughout the simulation.

Forces are deduced from the state (position **x**, velocity **v**) of the VE at time step t. Integration schemes compute the new state at $t + 1$ from the forces at t, with a time interval of Δt between t and $t + 1$.

Except for simple "school" cases, it is very difficult to have an analytical expression of $\mathbf{f}(\mathbf{x}, \mathbf{v})$ and to integrate it. Thus, integration schemes rely on approximating the differential equations and using numerical methods to solve them, yielding more or less approximated results depending on the type of scheme.

There are two main types of integration schemes: explicit and implicit. In explicit schemes, known quantities at time t are used to extrapolate quantities at time $t + 1$, assuming know quantities remain constant between two evaluations. This allows to treat every quantity independently, but does not guarantee the overall stability of the system. Explicit schemes, on the other hand, treat the quantities as coupled, solving them together at each time step as a system. This leads to an unconditionally stable system, where the system will remain stable no matter the time step Δt, but has a higher computational cost.

A.4.1 Explicit integration

There are many explicit integration schemes. The most commonly used in the haptic rendering literature are the explicit Euler method and the Runge-Kutta method.

A.4.1.1 The explicit Euler method

Euler's explicit integration method is the simplest way to integrate a function over an interval.

From eq. A.42, the global acceleration value at time $t+1$ is computed as:

$$\dot{\mathbf{v}}^{t+1} = M^{-1}\mathbf{f}(\mathbf{x}^t, \mathbf{v}^t) \tag{A.43}$$

Then, new velocities and positions at $t+1$ are integrated using their first order derivative, and by multiplying the derivative with the time step Δt:

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \Delta t \dot{\mathbf{v}}^t \tag{A.44}$$

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta t \mathbf{v}^t \tag{A.45}$$

This method is a first order approximation of the exact result (the error increases with the square of Δt). Due to the errors generated along the integrations, the size of the time step Δt must be chosen carefully in order to avoid the system to become unstable. In addition, whatever the time step used, the system diverges due to the accumulation of these errors.

A.4.1.2 The Runge-Kutta method

In Euler's explicit integration, the acceleration is evaluated once, at time t, and its value is integrated twice, leading to a potentially large error if the acceleration varies between times t and $t+1$.

The Runge-Kutta method reduces the integration error by evaluating forces several times at intermediate points, thus computing several accelerations values.

We will describe the Runge-Kutta stepping of order 2 (also called the mid-point method), but the order can be chosen arbitrarily. The higher the order of the Runge-Kutta step, the higher the precision of the integration, but also the higher the computational cost.

Let $\mathbf{x}^e$ be the position at time $t+1$ computed using Euler's explicit integration (eqs. A.43 to A.45). The Runge-Kutta method proposes to take into account the variation of the acceleration (coming from the variation of forces) by evaluating it at intermediate points. At order 2, a middle point $\mathbf{x}^{mid}$ is computed with:

$$\mathbf{x}^{mid} = \frac{\mathbf{x}^t + \mathbf{x}^e}{2} \tag{A.46}$$

And the acceleration is computed for the point $\mathbf{x}^{mid}$ as in (A.43):

$$\dot{\mathbf{v}}^{mid} = M^{-1}\mathbf{f}(\mathbf{x}^{mid}, \mathbf{v}^{mid}) \tag{A.47}$$

This intermediate acceleration is used at time $t+1$ to get $\mathbf{v}^{t+1}$ and $\mathbf{x}^{t+1}$:

$$\begin{aligned} \mathbf{v}^{t+1} &= \mathbf{v}^{t} + \Delta t \dot{\mathbf{v}}^{mid} && \text{(A.48)} \\ \mathbf{x}^{t+1} &= \mathbf{x}^{t} + \delta t \mathbf{v}^{t} && \text{(A.49)} \end{aligned}$$

Precision is greatly improved compared to Euler explicit integrations. The error is further reduced using higher orders, but requires the additional evaluation of forces at other intermediate points.

A.4.2 The implicit Euler method

The Euler and Runge-Kutta methods previously described are *explicit* methods because they are formulated on known quantities at time t, and extrapolate the final value at time $t+1$. Implicit methods propose to find the quantities at time $t+1$ by solving an "implicit" formulation that couples the integration outcome.

The most common implicit integration scheme is Euler's implicit integration method, which slightly modifies the formulation of Euler's explicit integration with:

$$\begin{aligned} \mathbf{v}^{t+1} &= \mathbf{v}^{t} + \Delta t \dot{\mathbf{v}}^{t+1} && \text{(A.50)} \\ \mathbf{x}^{t+1} &= \mathbf{x}^{t} + \Delta t \mathbf{v}^{t+1} && \text{(A.51)} \end{aligned}$$

In eq. A.50, the new velocity is computed from the acceleration at time $t+1$ (which is unknown since $\mathbf{x}^{t+1}$ and $\mathbf{v}^{t+1}$ are unknown), while in A.51, the new velocity must be known in order to compute the new state at time $t+1$. This system formulates the fact that we want to find a point $\mathbf{x}^{t+1}$ such that running the simulation backwards using explicit integration would give back point $\mathbf{x}^{t}$ after a time Δt (namely, from (A.51), $\mathbf{x}^{t} = \mathbf{x}^{t+1} - \Delta t \mathbf{v}^{t+1}$).

Since $\dot{\mathbf{v}}^{t+1}$ depends directly on the force function $\mathbf{f}(\mathbf{x}, \mathbf{v})$, we can evaluate $\dot{\mathbf{v}}^{t+1}$ using the first (or more) order approximation of $\mathbf{f}(\mathbf{x}, \mathbf{v})$. However, getting the value of $\mathbf{f}^{t+1}$ implies having the values of $\mathbf{x}^{t+1}$ and $\mathbf{v}^{t+1}$. Thus, it is possible to use the first order approximation of $\mathbf{x}^{t+1}$ and $\mathbf{v}^{t+1}$ to form a linear system that formulates the unknown $\dot{\mathbf{v}}^{t+1}$ as a function of $\mathbf{x}^{t}$, $\mathbf{v}^{t}$ and the force spatial derivatives (force Jacobian) as explained in the remainder of this section.

The force function $\mathbf{f}(\mathbf{x}, \mathbf{v})$ allowing to compute the acceleration is not necessarily linear on $\mathbf{x}$, leading to a non linear equation (A.50). In order to simplify the resolution of the system, we solve for $\dot{\mathbf{v}}^{t+1}$ using the first order Taylor expansion of the force function:

$$\mathbf{f}(\mathbf{x}^{t+1}, \mathbf{v}^{t+1}) = \mathbf{f}(\mathbf{x}^{t} + \Delta\mathbf{x}, \mathbf{v}^{t} + \Delta\mathbf{v}) = \mathbf{f}(\mathbf{x}^{t}, \mathbf{v}^{t}) + \Delta\mathbf{x}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \Delta\mathbf{v}\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \tag{A.52}$$

The acceleration $\dot{\mathbf{v}}$ at time $t+1$ becomes:

$$\dot{\mathbf{v}}^{t+1} = M^{-1}\left(\mathbf{f}(\mathbf{x}^{t}, \mathbf{v}^{t}) + \Delta\mathbf{x}\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \Delta\mathbf{v}\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right) \tag{A.53}$$

Using $\Delta\mathbf{x} = \Delta t(\mathbf{v}^{t} + \Delta t \dot{\mathbf{v}}^{t+1})$ and $\Delta\mathbf{v} = \Delta t \dot{\mathbf{v}}^{t+1}$, we obtain:

$$\left(I - M^{-1}\Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{v}} - M^{-1}(\Delta t)^{2}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\dot{\mathbf{v}}^{t+1} = M^{-1}\left(\mathbf{f}(\mathbf{x}^{t}, \mathbf{v}^{t}) + \Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{v}^{t}\right) \tag{A.54}$$

Using eq. A.54, we can solve for $\dot{\mathbf{v}}^{t+1}$ using an appropriate linear solver, such as the Conjugate Gradient method or the Gauss iterative method, among others. Once $\dot{\mathbf{v}}^{t+1}$ is known, $\mathbf{v}^{t+1}$ and $\mathbf{x}^{t+1}$ can be immediately computed using eqs. A.50 and A.51 respectively.

Appendix: Résumé Long en Français

B

Contents

B.1 Partie 1: Interaction Haptique et Multimodale avec des Milieux Complexes Basés Physique . . . 169
- B.1.1 Interaction haptique à 6DDL avec des fluides . . . 169
- B.1.2 Interaction haptique à 6DDL avec les différents états de la matière . . . 171
- B.1.3 Interaction vibrotactile et multimodale avec des fluides . . . 173

B.2 Partie 2: Navigation Immersive et Infinie Basée sur la Marche dans des Espaces de Travail Restreints . . . 176
- B.2.1 Navigation infinie d'EV dans des espaces de travail restreints en translation 176
- B.2.2 Navigation infinie d'EV dans des espaces de travail restreints en translation et en rotation . . . 177

B.3 Conclusion . . . 179

Dans ce manuscrit de thèse, nous présentons nos travaux de recherche conduits dans le contexte de la Réalité Virtuelle (RV). Les technologies de la RV visent à simuler des environnements numériques avec lesquels l'utilisateur peut interagir et, en conséquence, percevoir à travers différentes modalités les effets de ses actions en temps réel. Burdea et Coiffet [1] définissent la RV comme *"une interface utilisateur-ordinateur haut de gamme impliquant la simulation et l'interaction temps réel à travers plusieurs canaux sensoriels. Ces modalités sensorielles sont la vue, le toucher, l'audition, l'odorat et le goût"*.

Malheureusement, l'état de l'art montre plusieurs limitations en termes d'interaction, concernant les dispositifs matériels tout comme les composants logiciels. Plusieurs de ces limitations surgissent lors de l'interaction avec des environnements virtuels (EV) complexes. Par exemple, simuler des phénomènes naturels en respectant les contraintes de la RV, à savoir en temps réel et avec des retours sensoriels de haute qualité, est une tâche difficile. Il est aussi compliqué de concevoir un dispositif qui permette à l'utilisateur de marcher sans mouvement en avant tout en générant une véritable sensation de marche. Les facteurs limitants sont la puissance de calcul à disposition, la technologie limitée, et la complexité inhérente aux phénomènes naturels. En fait, la plupart des situations que nous vivons au quotidien ne peuvent pas être simulées de façon fidèle. Par exemple, il n'est pas encore possible de simuler fidèlement l'exploration multimodale de scènes naturelles, comme marcher sur une plage. Le mouvement de l'eau et l'enfoncement du sable sont deux phénomènes complexes, et même s'il existe des modèles physiques pour les simuler, les contraintes temps réel et les différents retours sensoriels constituent des défis considérables. Marcher naturellement dans l'EV afin d'atteindre l'autre bout de la plage est tout simplement impossible, en raison des limites dans l'espace de suivi de mouvement et de la minuscule taille des environnements immersifs comparé à la vraie taille d'un paysage naturel.

Contexte de Recherche

La RV est un outil puissant et immersif en grande partie grâce à sa capacité d'interaction avec l'utilisateur. Plus l'interaction et ses retours sensoriels sont crédibles, plus l'utilisateur est immergé dans l'EV. En considérant les capacités des systèmes de RV d'aujourd'hui et de demain, nous définissons notre contexte de recherche en nous appuyant sur trois conditions fondamentales pour l'interaction [1]. Ce contexte nous permettra de souligner les principales faiblesses des approches actuelles, et de montrer quels aspects sont restés largement inexplorés. Les problématiques qui en découlent nous permettront de définir nos différents axes de recherche et guideront ainsi notre travail.

- **L'interaction en RV doit être multimodale**. Dans le monde réel, nous interagissons avec notre environnement à travers nos sens. Chaque sens fourni des indices complémentaires pour une perception plus large et plus précise. Idéalement, cela devrait être aussi le cas dans les mondes virtuels.
- **L'interaction en RV doit être basée physique**. Le utilisateurs s'attendent à ce que l'EV réagisse comme dans le monde réel (à l'exception de quelques scénarios spécifiques): les objets doivent suivre les lois de la physique. Pour une interaction réaliste, le comportement doit être décrit par des modèles basés physique.
- **L'interaction en RV doit permettre des environnements complexes**. La plupart des situations dans le monde réel mettent en jeu des environnements complexes. Par complexe, nous faisons allusion à une plus grande demande dans les caractéristiques de l'EV et/ou des ses objets. Un EV peut être complexe de plusieurs façons: en taille (grandes scènes et objets), en nombre (grand nombre de polygones et d'objets), en forme (détails géométriques, objets convexes, relief de la scène), en comportement (milieux non-rigides, grandes forces, vitesses élevées), etc.

Pour satisfaire ces trois conditions, il faut faire face aux problèmes énoncés préalablement, à savoir une puissance de calcul limitée, une technologie limitée, et la complexité des phénomènes physiques naturels. Dans ce manuscrit, nous nous concentrons donc dans l'**amélioration des interactions multimodales et basées physique avec des EV complexes**.

Afin de mieux aborder cette problématique, nous la subdivisions en deux axes de recherche, suivant les principales catégories des techniques d'interaction en RV. Comme défini par Hinckley *et al.* [11], *"une technique d'interaction est la fusion d'entrées et de sorties, regroupant les éléments logiciels et matériels qui fournissent à l'utilisateur un moyen d'accomplir une tâche"*. A partir de la taxonomie de Bowman *et al.* [12] pour les tâches en RV, nous nous concentrons sur les deux principales catégories:

- la *manipulation* (et la *sélection*, étroitement liée), regroupant les techniques d'interaction permettant à l'utilisateur d'interagir avec les objets de l'EV,
- la *navigation*, regroupant les techniques d'interaction permettant à l'utilisateur de se déplacer dans l'EV.

---

[1] Ces conditions ne sont pas exclusives. Plusieurs utilisations de la RV ne demandent pas à ce que ces trois conditions soient respectées.

Ces catégories représentent les tâches qui peuvent être réalisées par un utilisateur dans le monde réel.

Objectifs et Contributions

Ce manuscrit est naturellement divisé en deux parties, chacune suivant un axe de recherche: d'une part, la *manipulation* multimodale d'EV complexes basés physique, et d'une autre part, la *navigation* multimodale d'EV complexes basés physique.

Dans le premier axe, nous nous concentrons sur l'**interaction haptique et multimodale avec des milieux complexes basés physique**. En effet, l'ajout de retours haptiques (forces, vibrations) permet d'augmenter considérablement l'immersion et la performance des utilisateurs dans des EV [14, 15]. Nous utilisons le sens du toucher dans quasiment toutes les tâches que nous accomplissons dans le monde réel. Cependant, les milieux complexes comme les fluides et les objets déformables n'ont pas été suffisamment étudiés dans ce contexte, malgré leur omniprésence dans la vie courante, la médecine et l'industrie.

Dans le deuxième axe, nous nous concentrons sur la **navigation immersive et infinie basée sur la marche dans des espaces de travail restreints**. En effet, utiliser la marche comme moyen de navigation permet non seulement d'améliorer les performances d'un utilisateur vis-à-vis d'une tâche virtuelle à effectuer [20, 21, 22], mais augmente aussi le sentiment de présence [23] et de naturalité [24, 23, 22] au sein de l'EV. Cependant, les espaces de travail courants de la RV (CAVE, zone de suivi de mouvements) sont restreints, présentant des limites qui peuvent être facilement atteintes par l'utilisateur, provoquant des erreurs, des blocages, ou des problèmes de sécurité.

Nous détaillons nos contributions par la suite, suivant les deux parties mentionnées préalablement.

Partie 1 - Interaction Haptique et Multimodale avec des Milieux Complexes Basés Physique

Dans cette première partie, nous proposons tout d'abord **une nouvelle approche pour l'interaction haptique à 6 degrés de liberté (DDL) avec des fluides**. L'interaction haptique avec des fluides pose plusieurs problèmes pour obtenir des retours de force réalistes et stables, compte tenu de la complexité de la simulation et des contraintes haute fréquence propres à l'haptique. Les travaux précédents proposent le pré-calcul d'une partie des forces [25], limitant ainsi les capacités d'interaction, des approches avec seulement 3DDL et des fluides non visqueux [26], ou des implémentations restreintes à des objets simples et de petites quantités de fluide [27]. Dans notre approche, nous proposons d'exploiter les 6DDL et d'interagir avec des fluides visqueux par le biais d'objets rigides de formes arbitraires, incluant des objets concaves pouvant transporter le fluide, permettant ainsi une interaction plus riche et réaliste. Nous utilisons un modèle basé particules, les Smoothed-Particle Hydrodynamics (SPH) [28, 29], une approche unifiée pour les corps rigides et les fluides, un nouveau schéma de couplage haptique et un rendu visuel adapté aux contraintes haptiques.

Nous proposons ensuite **une approche unifiée pour l'interaction haptique à 6DDL avec les différents états de la matière**. La complexité d'une simulation augmente considérablement lors de l'ajout d'objets déformables à une scène contenant déjà des fluides et des objets rigides. Chaque état de la matière (fluide, déformable, rigide) est régis par son propre modèle: ceux-ci doivent coexister en parallèle, et interagir entre eux.

Les couplages haptiques doivent prendre en compte les spécificités de chaque modèle. Ceci rajoute des algorithmes spécifiques et une complexité accrue à une simulation déjà complexe. A travers notre approche, nous proposons un mécanisme unifié pour l'interaction haptique à 6DDL avec ces différents états. Grâce à la possibilité de simuler des objets déformables avec les SPH [138, 151], nous utilisons notre approche pour les fluides et les objets rigides afin d'obtenir un couplage haptique passant par un seul schéma générique, transparent pour l'utilisateur. Notre approche est évaluée en mesurant la capacité des utilisateurs à reconnaître les états de la matière avec lesquels ils interagissent.

Enfin, nous proposons **un nouveau modèle pour l'interaction vibrotactile et multimodale avec des fluides**. Le retour de force et visuel ne sont pas les seules modalités importantes en RV. Les retours vibrotactile et acoustique fournissent des indices complémentaires pour une perception accrue des propriétés des matériaux, des forces et des distances, entre autres, tout en utilisant des dispositifs à bas coût et génériques. Des matériaux comme l'eau ou d'autres fluides n'ont pas été étudiés dans ce contexte. Notre approche permet donc d'interagir avec des fluides en utilisant les modalités vibrotactile, acoustique et, grâce à nos travaux précédents, visuelle et kinesthésique. Comme d'autres approches vibrotactiles [31, 32, 30], nous nous appuyons sur des modèles physiques de synthèse de son pour modéliser le retour vibrotactile, car les deux phénomènes partagent la même source physique. Nous explorons le retour multimodal à travers les mains et, de façon plus innovante, à travers les pieds lors de la marche sur des plages et des bassins virtuels.

Partie 2 - Navigation Immersive et Infinie Basée sur la Marche dans des Espaces de Travail Restreints

Dans cette deuxième partie, nous proposons tout d'abord **une nouvelle technique pour la navigation infinie d'EV dans des espaces de travail restreints en translation**. Lors de la navigation dans des EV plus larges que l'espaces de travail, l'utilisateur peut atteindre les limites de l'espace de travail lors de ses déplacements. Les solutions matérielles telles que les interfaces de locomotion [33] présentent des contraintes de taille, poids, coût et précision. Les solutions logicielles, les techniques de navigation [10, 34, 35, 36], ne fournissent pas de solution simple, intuitive et immersive. Notre approche vise donc à résoudre ces problèmes grâce à une nouvelle métaphore appelée le "Bandeau Magique", utilisant un contrôle hybride en position/vitesse basé sur la marche à l'intérieur de l'espace de travail et une loi de contrôle aux limites. Ces limites sont montrés explicitement à l'utilisateur au sein de l'EV sous la forme d'un bandeau. Deux expériences ont été menées afin d'évaluer notre approche par rapport à d'autres techniques répondant à la même problématique.

Nous proposons ensuite de **nouvelles techniques pour la navigation infinie d'EV dans des espaces de travail restreints en translation et en rotation**. Les limites des espaces de travail ne sont pas uniquement en translation: certains environnements, comme un CAVE, n'offrent pas un rendu visuel en 360°. Si l'utilisateur fait face à ces "écrans manquants", il s'expose à une rupture d'immersion. Certains espaces de travail présentent donc des limites en translation *et* en rotation. Nous proposons donc trois métaphores gérant ces limites, tout en utilisant la marche comme navigation principale. La première métaphore étend le paradigme du contrôleur (joystick) en rajoutant des signes virtuels pour prévenir l'utilisateur de sa proximité aux limites. La deuxième métaphore étend le Bandeau Magique en rajoutant des murs virtuels afin d'éviter que l'utilisateur se tourne vers le mur manquant. La troisième métaphore introduis un Compagnon Virtuel sous la forme d'un oiseau pour guider et protéger l'utilisateur dans l'EV. Ces techniques sont évaluées en les comparant entre elles et à l'utilisation simple d'un contrôleur.

Enfin, des conclusions et des perspectives sont présentées à la fin du manuscrit.

B.1 Partie 1: Interaction Haptique et Multimodale avec des Milieux Complexes Basés Physique

Dans cette première partie, nous abordons la *manipulation* multimodale d'EV complexes basés physique. Plus précisément, nous nous concentrons sur l'interaction haptique et multimodale avec des milieux complexes basés physique, tels que les fluides, les objets déformables, et les objets rigides.

B.1.1 Interaction haptique à 6DDL avec des fluides

Les fluides sont souvent présents dans les scénarios industriels, médicaux, et de divertissement, tout comme dans notre vie quotidienne. Permettre l'interaction haptique avec des fluides, sans contraintes majeures en terme de volume ou d'outil de manipulation comme jusqu'à présent, représente un apport majeur aux possibilités offertes par la RV.

Dans cette section, nous proposons la première approche permettant l'interaction haptique avec des fluides visqueux par le biais d'objets de forme arbitraire et d'interfaces haptiques à 6DDL.

B.1.1.1 Simulation physique de fluides

Afin de simuler les fluides, nous utilisons le modèle Smoothed-Particle Hydrodynamics (SPH) [28]. Une simulation SPH utilise un ensemble de particules portant différentes propriétés physiques, comme la masse et la viscosité, discrétisant ainsi le volume de fluide. Différentes quantités physiques, telles que la densité ou des forces d'interaction, peuvent être calculées pour chaque point de l'espace par l'interpolation des quantités portées par les particules voisines. Le voisinage est défini par le rayon d'interaction des particules, et l'interpolation par une fonction noyau.

Dans le contexte des fluides, le mouvement de l'ensemble des particules est régis par les equations de Navier-Stokes, mettant en jeu des forces de pression, de viscosité et de gravité. La simulation de fluides est effectuée en trois étapes [29]. Tout d'abord, la densité de chaque particule est calculée pour en déduire la pression. Ensuite, les forces de pression et de viscosité sont obtenues à partir des quantités calculées précédemment. Enfin, une nouvelle accélération, vitesse et position sont obtenues en utilisant les forces de l'étape précédente et une intégration d'Euler explicite.

Des paramètres tels que la viscosité peuvent être modifiés afin d'obtenir différents comportements du fluide, allant d'un gaz (pas de viscosité) au miel (très forte viscosité).

B.1.1.2 Simulation d'objets rigides

Afin de fournir un retour haptique à 6DDL, nous allons interagir avec les fluides par le biais d'un objet rigide. Nous avons donc besoin d'un modèle pour simuler des objets rigides qui s'adapte au modèle SPH. Nous proposons une approche temps-réel basée sur les travaux de Solenthaler *et al.* [138] et améliorée en augmentant ses performances pour de l'haptique. Il s'agît d'un modèle unifié pour la simulation de fluides et d'objets rigides.

Les objets rigides peuvent être simplement et efficacement modélisés avec les mêmes particules SPH que les fluides. Ces particules sont “gelées” afin que les positions relatives restent constantes, tout en appliquant la dynamique des objets rigides à l'ensemble de l'objet. Ceci permet l'utilisation du modèle SPH pour le calcul de forces entre fluides et objets rigides, sans avoir recours à des techniques de détection de collision supplémentaires. De plus, comme il s'agît d'une méthode basée particules, la forme des objets rigides n'est pas importante, et des objets concaves tels que des conteneurs peuvent être utilisés. Les maillages polygonaux sont donc convertis en un ensemble de particules, comme le montre la Figure B.1.

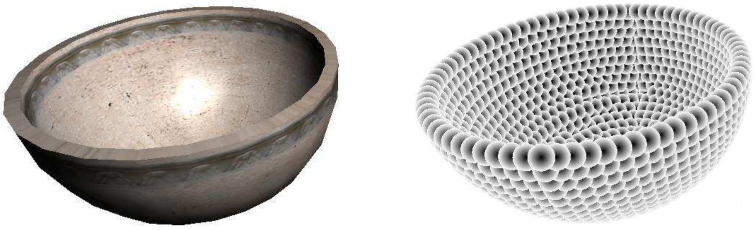


Figure B.1 – Transformation d'un maillage surfacique (gauche) en un ensemble de particules (droite)

B.1.1.3 Rendu haptique à 6DDL

Afin de calculer les forces à retourner à l'utilisateur, nous calculons tout d'abord les forces exercées par le fluide sur l'objet manipulé (le proxy) couplé à l'interface haptique. Dans un modèle unifié et avec des contraintes de temps critiques, nous proposons d'utiliser des forces d'interaction rigide-fluide unifiées. Nous utilisons donc les même forces d'interaction que dans un cas fluide-fluide, ce qui suppose un gain de calcul important en traitant toutes les particules de la même façon. Cette approche permet un contrôle raisonnable sur les forces d'interaction à travers des paramètres tels que la densité et la viscosité. Ces valeurs sont fixées en début de simulation pour chaque objet rigide afin d'obtenir des comportements différents.

Ainsi, l'algorithme de rendu haptique peut être décomposé en quatre étapes: 1) calcul de la densité pour chaque particule, 2) calcul des forces d'interaction pour chaque particule, 3) intégration des particules fluides, afin d'obtenir leur nouvelle vitesse et position, et 4) intégration et couplage des particules rigides, additionnant les forces d'interaction exercées sur chaque particule rigide afin d'obtenir une force et un couple total exercés sur l'objet en entier, et en tenant compte des forces fournies par l'interface haptique.

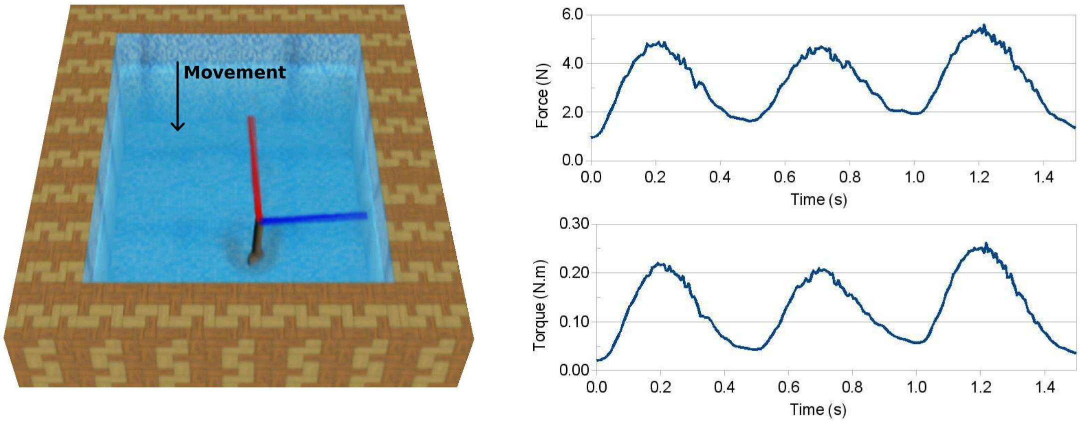
Un mécanisme de Couplage Virtuel [58] est introduit entre l'interface haptique et l'objet rigide, créant ainsi un lien visco-élastique afin d'augmenter la stabilité du système.

Ce schéma permet le couplage haptique avec n'importe quel objet rigide tant que sa masse et ses dimensions sont compatibles avec la portée et les efforts maximums autorisés par l'interface haptique. Ce schéma permet aussi le calcul de N couplages haptiques au sein du même EV, ainsi que le couplage entre N interfaces haptiques et le même objet rigide (comme en portant un saut à deux mains).

Ces différents algorithmes ont été implémentés sur GPU en utilisant CUDA, tirant ainsi parti de la nature parallèle de notre approche. Cette implémentation permet d'atteindre des fréquences suffisamment élevées pour la manipulation de fluides avec retour haptique (des fréquences entre 50 et 100 Hz sont satisfaisantes selon les travaux précédents), tout en interagissant avec des EV comportant de gros volumes de fluide.

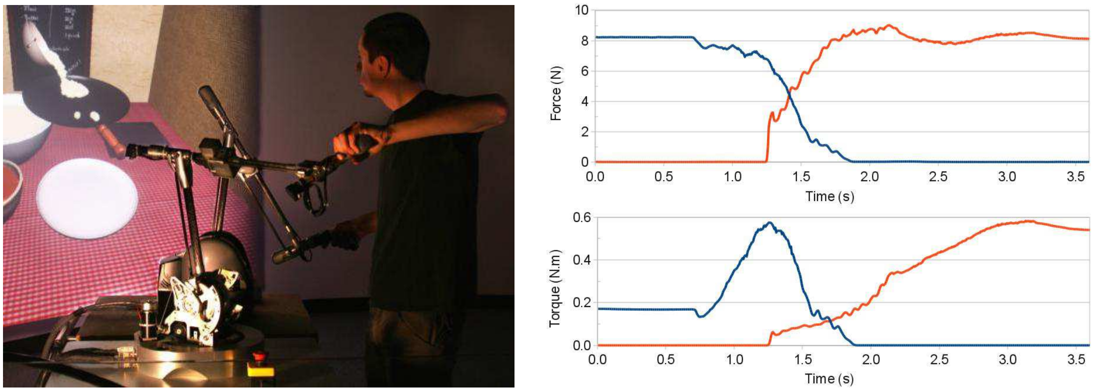
B.1.1.4 Scénarios de test

Nous avons conçu différents exemples d'interaction haptique avec des fluides afin de valider et d'illustrer les avantages de notre approche. Nous avons développé plusieurs scénarios, comme l'interaction à 6DDL (Figure B.2), l'interaction par le biais de conteneurs (Figure B.3), des cas de viscosité variable, le couplage bi-manuel sur un même objet et un exemple ludique de préparation de crêpes (Figure B.3, gauche).



(a) (b)

Figure B.2 – Interaction à 6DDL: une cuillère virtuelle est utilisée pour mélanger un fluide. (a) Force (en rouge) et couple (en bleu) exercés sur la cuillère. La force et le couple s'opposent au mouvement de la cuillère. (b) Graphique de la force (haut) et du couple (bas) générés pas le mouvement de la cuillère en avant, en arrière, puis encore en avant.



(a) (b)

Figure B.3 – Interaction par le biais de conteneurs. Un bol et une poêle, chacun couplé à une interface haptique à 6DDL, sont utilisés comme conteneurs. (a) Du fluide est versé du bol vers la poêle, faisant varier le poids d'une main à l'autre. (b) Graphique des forces (haut) et des couples (bas) du bol (en bleu) et de la poêle (en rouge) pendant le transvasement.

B.1.2 Interaction haptique à 6DDL avec les différents états de la matière

Les EV complexes ne sont pas restreints à un seul type de milieu. Plus généralement parlant, les EV complexes comportent plusieurs états de la matière simultanément. Dans la communauté graphique, les milieux sont définis par trois états de la matière: les objets rigides, les objets déformables, et les fluides. L'interaction haptique simultanée avec des milieux dans différents états présente des défis en termes de puissance de calcul requise et le besoin d'algorithmes de couplage physiques et haptiques spécifiques.

Dans cette section, nous nous penchons sur ce problème en présentant la première approche multi-état pour le rendu haptique, permettant l'interaction haptique simultanée avec des milieux à l'état fluide, déformable et rigide.

B.1.2.1 Simulation d'objets déformables

Les objets déformables suivent la mécanique des milieux continus et sa théorie de l'élasticité. Le modèle SPH peut être utilisé de façon très efficace pour simuler des objets déformables à partir des équations de la mécanique des milieux continus. Cette simulation peut se faire en tirant parti des structures de données et de la parallélisation sur GPU des calculs que nous avons mis en place pour simuler des fluides et des objets rigides. Nous avons donc porté sur GPU les travaux de Solenthaler *et al.* [138] pour le calcul de déformations non-linéaires, avec les modifications apportées par Becker *et al.* [151] pour rendre les calculs invariants à la rotation.

Nous avons aussi introduit un mécanisme de changement d'état, permettant des processus de solidification et de liquéfaction, en faisant varier les forces internes à chaque milieu (forces fluides, forces élastiques).

Nous avons utilisé deux GPUs dans cette approche: le premier effectue les calculs de simulation et de couplage haptique utilisant le modèle SPH, tandis que le second s'occupe de l'affichage graphique. Ceci permet d'utiliser un GPU en exclusivité pour la simulation et l'haptique, conduisant ainsi à un gain de performance de l'ordre de 400% par rapport à l'utilisation d'un seul GPU.

B.1.2.2 Rendu haptique multi-état à 6DDL

En rajoutant un troisième modèle physique permettant de simuler des objets déformables, les forces d'interaction unifiées présentées pour l'interaction haptique avec des fluides sont d'avantage mises à profit. Nous utilisons ces forces (pression et viscosité) pour l'interaction avec des objets déformables, gardant un couplage unique et générique entre les modèles, et présentant ainsi un avantage vis-à-vis du parallélisme des calculs: les mêmes forces sont calculées entre chaque particule appartenant à des objets différents.

Le schéma de couplage haptique reste lui aussi générique quel que soit l'état de la matière avec lequel l'utilisateur interagit, pour les mêmes raisons que le couplage entre modèles. L'utilisation de forces d'interaction unifiées nous affranchit de spécifier des couplages particuliers pour chaque état manipulé. De plus, grâce à l'introduction des objets déformables et des changements d'état, il est maintenant possible d'utiliser un proxy déformable pour interagir avec la matière (comme une cuillère en plastique pour mélanger un fluide visqueux), un proxy qui induit des changements d'état (par exemple, qui fait fondre la matière) ou qui subit des changements d'état (qui fond lui même).

B.1.2.3 Évaluation

Afin d'évaluer qualitativement notre approche haptique multi-état, nous avons mené une expérience perceptuelle avec 12 sujets dans laquelle ils devaient interagir avec les différents états de la matière. Notre objectif était d'évaluer la capacité des sujets à reconnaître les différents états (fluide, déformable, rigide) avec lesquels ils interagissaient, selon trois conditions: retour haptique seul, retour visuel seul, et les deux combinés. Les Figures B.4 et B.5 montrent les conditions expérimentales.

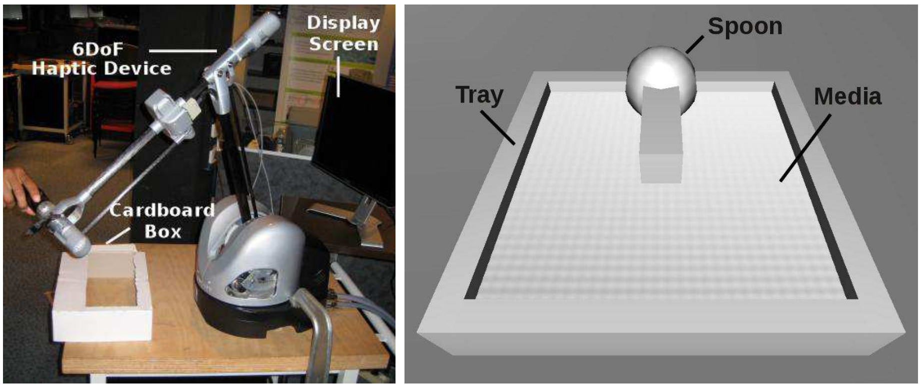


Figure B.4 – Conditions expérimentales: interfaces (gauche) et EV (droite) utilisés dans l'expérience.

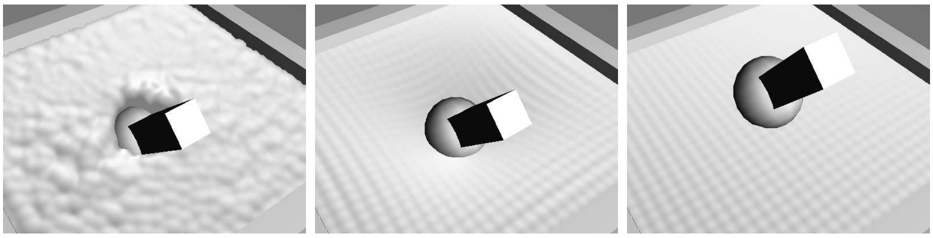


Figure B.5 – Les différents états de la matière. De gauche à droite: fluide, déformable, rigide.

Les résultats de l'expérience ont montré un taux de reconnaissance élevé (supérieur à 86%) pour les trois conditions et les trois états. Notre approche réussit donc à transmettre un retour de force en accord avec l'état simulé. Lorsque le rendu graphique est présent, le taux monte à 99% en moyenne pour les trois états. Cependant, le résultat le plus important est le score moyen de 87% de reconnaissance pour la condition haptique seule, montrant un rendu haptique de qualité. Les réponses des sujets à un questionnaire subjectif ont montré que l'ajout de retour haptique a amélioré le degré de réalisme et l'appréciation générale des participants.

B.1.3 Interaction vibrotactile et multimodale avec des fluides

Le retour vibrotactile permet d'apporter des indices supplémentaires à l'utilisateur concernant l'EV qui l'entoure, complémentaires aux retour de force et au retour visuel. C'est le cas de certaines informations haute fréquence, comme les textures, qui ne peuvent souvent pas être restitués par une interface haptique, et ne sont souvent pas visibles à l'œil nu. Même si le retour vibrotactile peut être généré pour plusieurs matériaux virtuels, l'eau et les fluides en général n'ont pas été étudiés dans ce contexte.

Dans cette section nous proposons donc le premier modèle basé physique pour le rendu vibrotactile d'interactions avec des fluides. Celui-ci s'appuie sur des connaissances dans la synthèse de sons de fluides, car les informations acoustiques et vibrotactiles partagent la même source physique. Les principales composantes sont donc l'impact solide-fluide à la surface, et le signal généré par les bulles en oscillation à l'intérieur.

B.1.3.1 Simulation SPH de bulles

Du fait que la majeure partie du son généré lors de l'interaction avec des fluides provienne des bulles à l'intérieur du fluide, nous avons étendu notre simulation de fluides SPH avec la capacité à générer des bulles d'air. Nous nous sommes inspirés des travaux de Müller *et al.* [143] pour développer un mécanisme basé physique pour détecter l'apparition d'une bulle d'air. Ce mécanisme utilise l'estimation de la quantité de particules de fluide autour d'une particule donnée, avec la direction vers laquelle se trouvent ces particules. Ainsi, quelques règles simples permettent de détecter la création d'une bulle: par exemple, si une particule fluide n'a pas suffisamment de voisins dans une direction donnée, c'est qu'elle est à côté d'une poche d'air, donnant ainsi naissance à une bulle.

B.1.3.2 Modèle vibrotactile

Le modèle s'articule autour de trois composantes, comme le montre la Figure B.6: l'impact initial, l'oscillation des bulles et l'oscillation des cavités. Les cavités sont formées par la pénétration du solide dans le fluide, créant un puits d'air qui se referme à la surface du fluide.

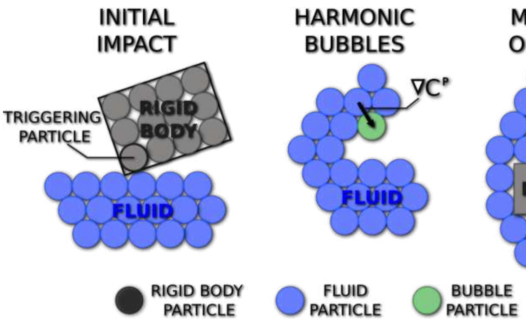


Figure B.6 – Les trois composantes de notre modèle vibrotactile.

Impact initial. Lors de l'impact d'un solide avec la surface du fluide, un son haute-fréquence, basse amplitude et très amortis peut être observé [153]. Nous exploitons la nature courte et “explosive” de cette vibration en la modélisant par la synthèse d'un signal de bruit blanc résonnant, pouvant contrôler sa fréquence de résonance, son amplitude et sa durée. Ce signal d'impact est généré lorsque la simulation détecte un contact entre la surface du solide et la surface du fluide. La fréquence et la durée sont prédéfinis, tandis que l'amplitude dépend de la vitesse v de l'objet rigide, en suivant le fait que l'intensité d'un tel impact est proportionnelle à v^3 [153].

Oscillation des bulles. Les bulles sont formées lorsque l'air est attrapé dans le fluide, dû à des mouvements à la surface ou à la fragmentation de cavités d'air sous la surface. La synthèse d'un signal vibrotactile se base sur les travaux de Minnaert[156], qui a proposé une approximation de la fréquence de résonance d'une bulle sphérique en fonction de son rayon. Nous utilisons donc cette relation et les apports de Van den Doel [155] pour une version temporelle (la fréquence augmente quand la bulle se rapproche de la surface) pour générer un signal correspondant à la vibration d'une bulle d'air. Ce signal est généré quand la simulation physique détecte la formation d'une bulle d'air. Le rayon de la bulle est déterminé suivant les observation de travaux antérieurs, modélisant la distribution des longueurs de rayon par une loi de puissance [158].

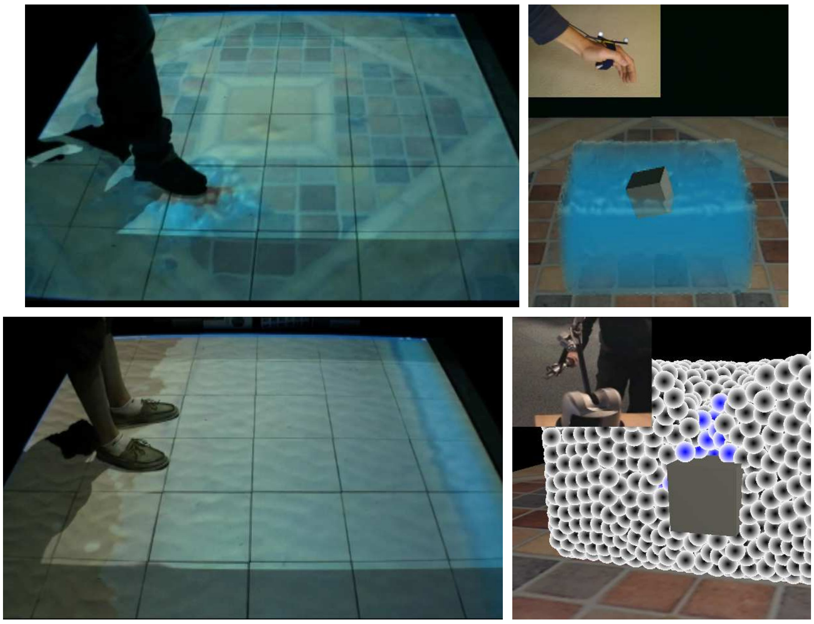


Figure B.7 – Exemples d'interaction: pied-fluide en marchant dans un bassin (gauche, haut), pied-fluide en sentant une vague sous les pieds (gauche, bas), main-fluide dans une bassine avec un vibreur (droite, haut) ou une interface haptique à 6DDL (droite, bas).

Oscillation des cavités. La pénétration d'un objet dans le volume de fluide forme une cavité d'air qui, en se refermant à la surface du fluide, génère une vibration basse-fréquence caractéristique, reconnaissable à l'écoute. Nous traitons cette cavité comme une grande bulle sphérique, et nous réutilisons donc notre composante pour l'oscillation de bulles afin de générer le son correspondant. Afin de détecter la cavité et de mesurer son volume, nous détectons le regroupement de particules d'air au sein du volume de fluide. Quand une particule d'air est entourée exclusivement d'autres bulles d'air, une cavité est potentiellement formée. En partant de la particule d'air détectée, nous procédons à une recherche de voisins afin de connaître l'étendue de la cavité et son volume. En considérant la cavité comme sphérique, nous en déduisons son rayon, et nous transmettons cette donnée à l'algorithme de génération de signal vibrotactile.

B.1.3.3 Rendu vibrotactile et multimodal

Nous avons développé plusieurs scénarios pour tester et valider notre approche, comme illustré Figure B.7. Nous avons utilisé de simples vibreurs attachés aux mains de l'utilisateur, ainsi qu'un sol composé de dalles vibrantes pour une restitution du signal par les pieds, simulant des scènes telles qu'un bassin (marche active de l'utilisateur) ou une plage (l'utilisateur reste sur place pendant qu'une vague passe sous ses pieds).

Afin d'obtenir un retour multimodal, d'autres modalités peuvent être rajoutés. Le rendu acoustique est possible en utilisant le même modèle, car les deux phénomènes proviennent de la même source physique. Pour cela, nous avons utilisé de simples haut parleurs. Le retour de force est aussi possible grâce à l'approche que nous avons proposé en section B.1.1.

Une étude utilisateur préliminaire, utilisant les dalles vibrantes et la scène du bassin, a rendu des résultats très encourageants, avec 6 sujets sur 8 ayant reconnu de l'eau sous leur pieds en n'utilisant que le retour vibrotactile.

B.2 Partie 2: Navigation Immersive et Infinie Basée sur la Marche dans des Espaces de Travail Restreints

Dans cette deuxième partie, nous abordons la *navigation* multimodale d'EV complexes basés physique. Plus précisément, nous nous concentrons sur la navigation immersive et infinie basée sur la marche dans des espaces de travail restreints, présentant des limites en translation (écrans physiques, espace de suivi de mouvements limité) et en rotation (écrans manquants).

B.2.1 Navigation infinie d'EV dans des espaces de travail restreints en translation

La navigation dans de grands EV au sein d'un espace de travail présentant des limites de translation est un vrai problème. Les utilisateurs finissent par atteindre les limites de l'espace de travail, soulevant des problèmes potentiels de collision, de perte de suivi de mouvement, et de rupture d'immersion.

Dans cette section, nous abordons ce problème en introduisant le "Bandeau Magique", une nouvelle métaphore d'interaction basée sur la marche pour la navigation dans des EV potentiellement infinis au sein d'un espace de travail restreint en translation.

B.2.1.1 Le "Bandeau Magique"

Le Bandeau Magique permet tout d'abord de montrer clairement les limites de l'espace de travail au sein de l'EV, et donc de façon immersive. Ces limites sont détourées par un bandeau prenant la forme de l'espace de travail, se matérialisant au fur et à mesure que l'utilisateur se rapproche des limites. Ainsi, si l'utilisateur reste à l'intérieur de la zone délimitée par le bandeau, il sera en sécurité. Le bandeau, situé à mi-hauteur, est texturé à la façon d'un ruban de balisage jaune et noir, évoquant le message implicite "ne pas passer". Un deuxième bandeau, situé à la hauteur de la tête de l'utilisateur, n'apparaît que quand celui-ci est très proche des limites. L'ombre des bandeaux est projetée par terre, afin de donner un indice visuel supplémentaire, comme le montre la Figure B.8.

Figure B.8 – Les indices visuels du Bandeau Magique pour montrer les limites de l'espace de travail.

Pour permettre la navigation au-delà des limites, nous utilisons une approche hybride de contrôle en position/vitesse [193]. L'utilisateur est en contrôle en position lorsque il marche à l'intérieur de l'espace de travail. Quand il arrive aux limites, il peut basculer en contrôle en vitesse en appuyant sur le bandeau (Figure B.9), permettant ainsi de translater son point de vue dans l'EV. Le bandeau se déforme, montrant à l'utilisateur que le contrôle en vitesse est activé. La loi de contrôle en vitesse est proportionnelle à la distance de pénétration dans le bandeau.

Figure B.9 – Le Bandeau Magique en utilisation: l'utilisateur (gauche) pousse le Bandeau Magique (milieu) pour se déplacer dans l'EV. N'importe quelle partie du corps avec suivi peut être utilisée, comme par exemple le coude (droite).

B.2.1.2 Évaluation

Afin d'évaluer l'efficacité de notre approche pour la navigation dans des espaces de travail restreints sans rupture d'immersion, nous avons mené une expérience en comparant notre approche à d'autres techniques de l'état de l'art répondant aux mêmes besoins [35]. Cette expérience comporte une tâche de pointage (l'utilisateur doit se déplacer à un endroit précis de l'EV) et une tâche de suivi de chemin. Douze sujets ont pris part à cette expérience.

Les résultats ont montré que le Bandeau Magique permet une navigation plus rapide et aisé par rapport aux autres techniques évaluées, tout en étant moins précise. Les sujets ont préféré notre approche sur la plupart des critères subjectifs évalués, tels que la naturalité, la fatigue et l'appréciation générale.

B.2.2 Navigation infinie d'EV dans des espaces de travail restreints en translation et en rotation

Certains environnements immersifs, comme les espaces type CAVE, permettent d'atteindre une très grand degré immersion dans les EV. Cependant, ces environnements présentent des limites en translation, comme observé préalablement, mais aussi en rotation. Le rendu visuel ne se faisant pas sur 360°, l'utilisateur peut regarder dans la direction des écrans "manquants", et ainsi subir une rupture d'immersion.

Dans cette section, nous présentons trois nouvelles métaphores proposant une solution à ces problèmes: le contrôleur contraint avec signes, le Bandeau Magique étendu, et le Compagnon Virtuel.

B.2.2.1 Trois nouvelles techniques de navigation

Nos trois nouvelles métaphores varient du plus simple au plus complexe, chacune étant un peu plus intégrée à l'EV. Elles sont illustrées Figure B.10. Les trois métaphores s'articulent autour de deux composantes: un mécanisme pour montrer les limites en translation et en rotation, et un mécanisme de navigation en contrôle hybride position/vitesse.

La première technique est une extension directe du paradigme du contrôleur. L'utilisateur ne peut utiliser le contrôleur que quand il est près des limites, le forçant ainsi à marcher.

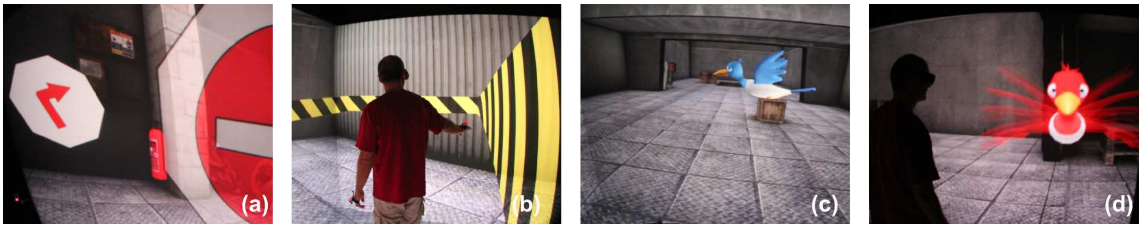


Figure B.10 – Photographies illustrant nos trois techniques. (a) Contrôleur contraint et signes: les signes “sens interdit” et “tourner à droite”. (b) Bandeau Magique étendu: le bandeau et les murs virtuels. (c,d) Le Compagnon Virtuel: l’oiseau au repos (c) et en protection (d).

Des signes de prévention (sens interdit, tourner à gauche, tourner à droite) s’affichent progressivement, alignés avec les bords de l’espace de travail en translation ou faisant face à l’utilisateur en rotation.

La deuxième technique étend le Bandeau Magique, présenté en section B.2.1, en lui rajoutant des murs virtuels (comme des œillères) pour les limites en rotation. Ces murs visent à empêcher l’utilisateur de tourner au-delà d’un certain angle, afin d’éviter que l’écran manquant rentre dans son champ de vision. Il peut se déplacer en translation et en rotation en poussant sur le bandeau et les murs virtuels, respectivement.

La troisième technique s’appuie sur un Compagnon Virtuel sous la forme d’un oiseau qui se déplace dans l’EV. L’oiseau empêche l’utilisateur d’atteindre les limites en s’interposant entre les deux, en translation (face à l’utilisateur) comme en rotation (venant du côté de l’écran manquant). A travers un ensemble de gestes et une laisse, l’utilisateur peut contrôler l’oiseau afin de se déplacer dans l’EV, comme s’il contrôlait un cheval. Les gestes sont décris Figure B.11.

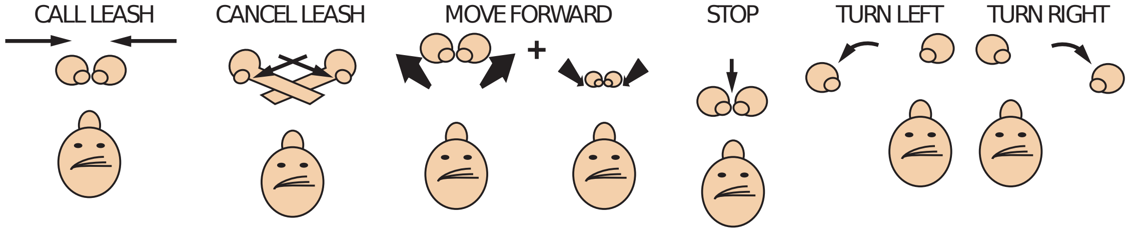


Figure B.11 – L’ensemble de gestes pour contrôler le Compagnon Virtuel et se déplacer dans l’EV.

B.2.2.2 Évaluation

Afin d’évaluer nos trois métaphores, nous les avons comparées entre elles et avec une condition de base (contrôleur non contraint) à travers une expérience utilisateur. Cette expérience comporte une tâche de pointage (l’utilisateur doit se déplacer à un endroit précis de l’EV) et une tâche de suivi de chemin. Douze sujets ont pris part à cette expérience.

Les résultats ont montré que les trois techniques remplissent leur objectifs, à savoir proposer une navigation tenant compte des limites. Le Compagnon Virtuel est l’approche la plus écologique, tandis que le Bandeau Magique étendu permet de mieux protéger l’utilisateur en rotation. Cet étude permet d’avoir un aperçu sur les compromis devant être effectués lors de la conception d’une technique de navigation répondant à ses besoins. Nous sommes particulièrement optimistes vis-à-vis du Compagnon Virtuel et de son potentiel pour des techniques de navigation futures.

B.3 Conclusion

Dans ce manuscrit de thèse, nous avons étudié la conception de retours multimodaux et de techniques d'interaction pour des EV larges et basés physique. L'objectif principal était d'améliorer l'interaction entre l'utilisateur et des EV complexes au sein de simulations de RV. Nous avons suivi deux axes de recherche, correspondant à deux catégories fondamentales d'interaction. Dans le premier axe (Partie 1), correspondant à la catégorie *manipulation*, nous nous sommes concentrés sur le rendu de retours haptiques et multimodaux lors de l'interaction avec des fluides, des objets déformables et des objets rigides. Dans le deuxième axe (Partie 2), correspondant à la catégorie *navigation*, nous nous sommes concentrés sur la navigation immersive basée sur la marche dans des EV potentiellement infinis au sein d'espaces de travail restreints.

Plus précisément, dans la première partie de ce manuscrit nous avons tout d'abord présenté une nouvelle approche pour l'interaction haptique à 6DDL avec des fluides. Cette approche permet la génération de retours de force lors de l'interaction avec des fluides visqueux par le biais d'objets rigides de forme arbitraire. Cette approche est étendue pour inclure l'interaction haptique avec des objets déformables, menant donc à une interaction haptique unifiée avec les différents états de la matière. Une expérience perceptuelle nous a permis de montrer que les utilisateurs pouvaient identifier de façon efficace les différents états en n'utilisant que la modalité haptique. Ensuite, nous avons introduit un nouveau modèle pour un retour vibrotactile lors de l'interaction avec des fluides, tirant parti de connaissances dans la synthèse de son de fluides. A travers cette approche, nous rendons possible l'interaction avec des fluides tout en générant des retours multimodaux, utilisant les canaux sensoriels vibrotactile, kinesthésique, acoustique et visuel.

Dans la seconde partie de ce manuscrit, nous avons présenté une nouvelle technique de navigation, nommée le Bandeau Magique, qui permet à l'utilisateur de connaître de façon immersive les limites de son espace de travail en translation grâce à une bandeau prenant la forme de ces limites. En utilisant un contrôle hybride en position/vitesse, cette technique fournis une métaphore simple et intuitive pour une navigation libre de collisions et de ruptures d'immersion. Comme certains espaces de travail présentent aussi des limites en rotation dues à des écrans manquants, comme dans le cas d'un CAVE, nous avons ensuite proposé trois nouvelles métaphores de navigation abordant ces problèmes: l'utilisation de signes visuels pour montrer les limites et d'un contrôleur contraint, l'extension du Bandeau Magique avec des murs virtuels pour les rotations, et un Compagnon Virtuel protégeant et guidant l'utilisateur dans l'EV. L'évaluation de ces techniques de navigation a permis de montrer qu'elles remplissent efficacement leurs objectifs tout en étant très appréciées par les utilisateurs.

Naturellement, beaucoup de travaux sont encore nécessaires afin d'obtenir une interaction multimodale complète avec n'importe quel EV. Grâce aux améliorations technologiques dont profitera la RV dans les années à venir, les prochaines décennies seront témoins d'énormes avancées dans ce domaine, tout en relevant de nouveaux et passionnants défis. Nous espérons que les travaux présentés dans ce manuscrit contribueront dans ce contexte, construisant le pont entre les mondes réels et virtuels.

Author's publications

Published

[P1] **Gabriel Cirio**, Maud Marchal, Tony Regia-Corte and Anatole Lécuyer. "The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace". In *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, pages 155–162, 2009

[P2] Maud Marchal, Anatole Lécuyer, **Gabriel Cirio**, Laurent Bonnet, Mathieu Emily. "Walking Up and Down in Immersive Virtual Worlds: Novel Interaction Techniques Based on Visual Feedback". In *Proceedings of IEEE Symposium on 3D User Interfaces*, pages 19–26, 2010

[P3] **Gabriel Cirio**, Maud Marchal, Aurélien Le Gentil and Anatole Lécuyer. "Tap, squeeze and stir the virtual world: Touching the different states of matter through 6DoF haptic interaction". In *Proceedings of IEEE Virtual Reality*, pages 123–126, 2011

[P4] **Gabriel Cirio**, Maud Marchal, Sébastien Hillaire and Anatole Lécuyer. "Six Degrees-of-Freedom Haptic Interaction with Fluids". *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1714–1727, 2011

[P5] **Gabriel Cirio**, Maud Marchal, Sébastien Hillaire and Anatole Lécuyer. "The Virtual Crepe Factory: 6DoF Haptic Interaction with Fluids". In *ACM SIGGRAPH Emerging Technologies*, 2011

[P6] Taku Hachisu, **Gabriel Cirio**, Maud Marchal, Anatole Lécuyer and Hiroyuki Kajimoto. "Virtual Chromatic Percussions Simulated by Pseudo-Haptic and Vibrotactile Feedback". In *Proceedings of ACM International Conference on Advances in Computer Entertainment Technology*, 2011, to appear.

Submitted

[S1] **Gabriel Cirio**, Maud Marchal, Anatole Lécuyer and Jeremy R. Cooperstock. "Vibrotactile Rendering of Fluids". Submitted to *IEEE Transactions on Haptics.*

[S2] **Gabriel Cirio**, Peter Vangorp, Maud Marchal, Emanuelle Chapoulie, Anatole Lécuyer and George Drettakis. "Walking in a Cube: Novel Metaphors for Safely Navigating Large Virtual Environments in Restricted Real Workspaces". Submitted to *IEEE Virtual Reality* 2012.

[S3] **Gabriel Cirio**, Maud Marchal and Anatole Lécuyer. "6DoF Haptic Rendering of Multi-State Simulations with Fluid, Deformable and Rigid Media". Submitted to *Computer & Graphics.*

[S4] **Gabriel Cirio**, Yon Visell, Maud Marchal and Anatole Lécuyer. "Multisensory and Haptic Rendering of Complex Virtual Grounds", Book Chapter of "Walking with the Senses: Non-visual perceptual techniques for walking in simulated environments", ed. Y. Visell, F. Fontana. Submitted to *Logos Verlag* .

Bibliography

[1] G. C. Burdea and P. Coiffet, *Virtual Reality Technology*, 2nd ed. Wiley-IEEE Press, Jun. 2003. 1, 165

[2] J. Sreng, F. Bergez, J. Legarrec, A. Lécuyer, and C. Andriot, “Using an event-based approach to improve the multimodal rendering of 6DOF virtual contact.” ACM Press, 2007, p. 165. 1, 29, 30, 41, 42

[3] A. Lecuyer, “Using eyes, hands, and brain for 3D interaction with virtual environments: A Perception-Based approach,” Habilitation Thesis, University of Rennes 1, 2010. 1

[4] S. Cotin, H. Delingette, and N. Ayache, “Real-Time elastic deformations of soft tissues for surgery simulation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999. 1, 32

[5] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot, “Realistic haptic rendering of interacting deformable objects in virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 1, pp. 36–47, 2006. 1, 32, 34

[6] D. Wang, Y. Zhang, and Z. Sun, “Multi-modal virtual reality dental training system with integrated Haptic-Visual-Audio display,” in *Robotic Welding, Intelligence and Automation*, ser. Lecture Notes in Control and Information Sciences, T. Tarn, S. Chen, and C. Zhou, Eds. Springer Berlin / Heidelberg, 2007, vol. 362, pp. 453–462, 10.1007/978-3-540-73374-4_54. 1, 42

[7] R. Riener, M. Frey, T. Proll, F. Regenfelder, and R. Burgkart, “Phantom-based multimodal interactions for medical education and training: the munich knee joint simulator,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, no. 2, pp. 208–216, 2004. 1, 42, 43

[8] H. Esen, K. Yano, and M. Buss, “A virtual environment medical training system for bone drilling with 3 DOF force feedback,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*, vol. 4. IEEE, 2004, pp. 3631– 3636. 1, 42

[9] M. Krijn, P. Emmelkamp, R. Biemond, de Wilde, M. Schuemie, and C. van der Mast, “Treatment of acrophobia in virtual reality: The role of immersion and presence,” *Behaviour Research and Therapy*, vol. 42, no. 2, pp. 229–239, 2004. 1

[10] M. Slater, M. Usoh, and A. Steed, “Taking steps: the influence of a walking technique on presence in virtual reality,” *ACM Trans. Comput.-Hum. Interact.*, vol. 2, no. 3, pp. 201–219, 1995. 2, 9, 103, 168

[11] K. Hinckley, R. J. Jacob, and C. Ware, *Computer Science Handbook*, 2nd ed., A. B. Tucker, Ed. Chapman and Hall/CRC, Jun. 2004. 3, 166

[12] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*, 1st ed. Addison-Wesley Professional, 2004. 3, 102, 166

[13] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Computers & Graphics*, vol. 21, no. 4, pp. 393–404, 1997. 4, 14, 15, 16, 17, 18, 19, 21, 30

[14] R. J. Adams, D. Klowden, and B. Hannaford, "Virtual training for a manual assembly task," *Haptics-e, The Electronic Journal of Haptics Research*, vol. 2, 2001. 4, 14, 38, 167

[15] S. Lee and G. Kim, "Effects of haptic feedback, stereoscopy, and image resolution on performance and presence in remote navigation," *International Journal of Human-Computer Studies*, vol. 66, pp. 701–717, 2008. 4, 14, 38, 45, 167

[16] Y. Visell, J. Cooperstock, and K. Franinovic, "The EcoTile: an architectural platform for Audio-Haptic simulation in walking," in *Proc. of the 4th Intl. Conf. on Enactive Interfaces*, Grenoble, France, 2007. 4, 35, 92

[17] S. Papetti, F. Fontana, M. Civolani, A. Berrezag, and V. Hayward, "Audio-tactile display of ground properties using interactive shoes," in *Haptic and Audio Interaction Design*, ser. Lecture Notes in Computer Science, R. Nordahl, S. Serafin, F. Fontana, and S. Brewster, Eds. Springer Berlin / Heidelberg, 2010, vol. 6306, pp. 117–128, 10.1007/978-3-642-15841-4_13. 4, 40, 84

[18] P. Richard, M. Nicodemi, R. Delannay, P. Ribière, and D. Bideau, "Slow relaxation and compaction of granular systems," *Nature Materials*, vol. 4, pp. 121–128, 2005. 4

[19] C. Cadoz, A. Luciani, and J. L. Florens, "CORDIS-ANIMA: modeling and simulation system for sound and image synthesis - the general formalism," *Computer Music Journal*, vol. 17, no. 1, 1993. 5, 39

[20] R. A. Ruddle and S. Lessels, "The benefits of using a walking interface to navigate virtual environments," *ACM Trans. Comput.-Hum. Interact.*, vol. 16, no. 1, pp. 1–18, 2009. 6, 95, 111, 128, 167

[21] M. Heintz, "Real walking in virtual learning environments: Beyond the advantage of naturalness," in *Learning in the Synergy of Multiple Disciplines*, U. Cress, V. Dimitrova, and M. Specht, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5794, pp. 584–595. 6, 95, 111, 128, 167

[22] C. A. Zanbaka, B. C. Lok, S. V. Babu, A. C. Ulinski, and L. F. Hodges, "Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp. 694–705, 2005. 6, 95, 111, 167

[23] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and J. Frederick P. Brooks, "Walking > walking-in-place > flying, in virtual environments," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 359–364. 6, 95, 103, 111, 128, 167

[24] W. L. Kuan and C. Y. San, “Constructivist physics learning in an immersive, multi-user hot air balloon simulation program (iHABS),” in *ACM SIGGRAPH 2003 Educators Program*. ACM Press, 2003, p. 1. 6, 95, 111, 167

[25] Y. Dobashi, M. Sato, S. Hasegawa, T. Yamamoto, M. Kato, and T. Nishita, “A fluid resistance map method for real-time haptic interaction with fluids,” in *Proceedings of the ACM symposium on Virtual reality software and technology*. Limassol, Cyprus: ACM, 2006, pp. 91–99. 7, 37, 38, 45, 62, 63, 167

[26] M. Yang, J. Lu, Z. Zhou, A. Safonova, and K. Kuchenbecker, “A GPU-Based approach for Real-Time haptic rendering of 3D fluids,” in *Proceedings of ACM SIGGRAPH Asia Sketches*, Dec. 2009, sketches. 7, 37, 39, 45, 57, 62, 167

[27] W. Baxter and M. C. Lin, “Haptic interaction with fluid media,” in *Proceedings of Graphics Interface 2004*. London, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 81–88. 7, 37, 39, 45, 57, 62, 167

[28] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Annual Review of Astronomy and Astrophysics*, vol. 30, no. 1, pp. 543–574, 1992. 7, 46, 167, 169

[29] M. Muller, D. Charypar, and M. Gross, “Particle-based fluid simulation for interactive applications,” in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. San Diego, California: Eurographics Association, 2003, pp. 154–159. 7, 46, 47, 167, 169

[30] R. Nordahl, A. Berrezag, S. Dimitrov, L. Turchet, V. Hayward, and S. Serafin, “Preliminary experiment combining virtual reality haptic shoes and audio synthesis,” in *Haptics: Generating and Perceiving Tangible Sensations*, ser. Lecture Notes in Computer Science, A. Kappers, J. van Erp, W. Bergmann Tiest, and F. van der Helm, Eds. Springer Berlin / Heidelberg, 2010, vol. 6192, pp. 123–129, 10.1007/978-3-642-14075-4_18. 8, 40, 83, 84, 168

[31] Y. Visell, J. Cooperstock, B. Giordano, K. Franinovic, A. Law, S. McAdams, K. Jathal, and F. Fontana, “A vibrotactile device for display of virtual ground materials in walking,” in *Haptics: Perception, Devices and Scenarios*, 2008, pp. 420–426. 8, 18, 19, 35, 40, 83, 84, 89, 146, 168

[32] Y. Visell, A. Law, J. Ip, S. Smith, and J. Cooperstock, “Interaction capture in immersive virtual environments via an intelligent floor surface,” in *Virtual Reality Conference (VR), 2010 IEEE*, 2010, pp. 313–314. 8, 35, 36, 40, 83, 84, 168

[33] J. Hollerbach, “Locomotion interfaces,” in *Handbook of Virtual Environments*, K. M. Stanney, Ed., 2002, p. 1232. 9, 95, 96, 99, 168

[34] J. J. LaViola, D. A. Feliz, D. F. Keefe, and R. C. Zeleznik, “Hands-free multi-scale navigation in virtual environments,” in *Proceedings of the ACM symposium on Interactive 3D graphics*. ACM, 2001, pp. 9–15. 9, 103, 104, 168

[35] B. Williams, G. Narasimham, B. Rump, T. P. McNamara, T. H. Carr, J. Rieser, and B. Bodenheimer, “Exploring large virtual environments with an HMD when physical space is limited,” in *Proceedings of the ACM symposium on Applied perception in graphics and visualization*. Tubingen, Germany: ACM, 2007, pp. 41–48. 9, 104, 115, 116, 168, 177

[36] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected walking," in *Proceedings of Eurographics*, 2001. 9, 105, 106, 128, 129, 139, 168

[37] C. Basdogan and M. A. Srinivasan, *Handbook of Virtual Environments: Design, Implementation, and Applications*, 1st ed., K. Stanney, Ed. CRC Press, Jan. 2002. 13, 19

[38] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: introductory concepts," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 24– 32, 2004. 13, 14, 15

[39] I. E. Sutherland, "The ultimate display," in *Proceedings of the IFIP Congress*, 1965, p. 506–508. 14

[40] J. Batter and F. Brooks, "Grope-1: a computer display to the sense of feel," in *nformation Processing, Proceedings of IFIP Congress*, 1971, pp. 759–763. 14

[41] A. M. Noll, "Man-Machine tactile communication," *Journal of the Society for Information Display*, vol. 1, no. 2, pp. 5–11, 1972. 14

[42] R. Viciana-Abad, A. R. Lecuona, and M. Poyade, "The influence of passive haptic feedback and difference interaction metaphors on presence and task performance," *Presence: Teleoperators and Virtual Environments*, vol. 19, no. 3, pp. 197–212, 2010. 14, 38, 45

[43] J. M. Loomis and S. J. Lederman, "Tactual perception," in *Engineering Data Compendium: Human Perception and Performance.* Harry G Armstrong Aerospace, 1988. 16

[44] H. Pongrac, "Vibrotactile perception: Differential effects of frequency, amplitude, and acceleration," in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications, 2006. HAVE 2006.* IEEE, 2006, pp. 54–59. 17

[45] L. A. Jones, "Kinesthetic sensing," in *Proceedings of Workshop on Human and Machine Haptics*, 1997. 17

[46] G. C. Burdea, *Force and Touch Feedback for Virtual Reality*, 1st ed. Wiley-Interscience, Aug. 1996. 17

[47] H. Z. Tan, M. A. Srinivasan, B. Eberman, and B. Cheng, "Human factors for the design of force-reflecting haptic interfaces," in *Proceedings of ASME Dynamic Systems and Control*, vol. 55, 1994, pp. 353–359. 17, 18

[48] J. Zhou, F. Malric, E. M. Petriu, and N. D. Georganas, "Uniform hardness perception in 6-DOF haptic rendering," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 214–225, 2011. 18

[49] S. D. Laycock and A. M. Day, "Recent developments and applications of haptic devices," *Computer Graphics Forum*, vol. 22, no. 2, pp. 117–132, Jun. 2003. 18, 19

[50] M. Lin and M. A. Otaduy, *Haptic Rendering: Foundations, Algorithms and Applications*, illustrated edition ed., M. C. Lin and M. Otaduy, Eds. A K Peters, Jul. 2008. 18, 35, 48, 54, 55, 57, 69

[51] C. R. Wagner, S. J. Lederman, and R. D. Howe, "A tactile shape display using RC servomotors," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings.* IEEE, 2002, pp. 354–355. 18, 19

[52] Q. Wang and V. Hayward, "Compact, portable, modular, high-performance, distributed tactile transducer device based on lateral skin deformation," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.* IEEE, 2006, pp. 67– 72. 18, 19

[53] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004. 18

[54] S. Zhai, "Human performance in six degree of freedom input control," Ph.D. dissertation, University of Toronto, 1995. 18, 119, 124

[55] Y. Hirata and M. Sato, "3-dimensional interface device for virtual work space," in *Proceedings of the lEEE/RSJ International Conference on Intelligent Robots and Systems, 1992*, vol. 2. IEEE, 1992, pp. 889–896. 19, 20

[56] C. R. Carignan and K. R. Cleary, "Closed-loop force control for haptic simulation of virtual environments," *HAPTICS-E*, vol. 1, pp. 01—1, 2000. 20, 21, 22

[57] J. E. Colgate, P. E. Grafing, M. C. Stanley, and G. Schenkel, "Implementation of stiff virtual walls in force-reflecting interfaces," in *, 1993 IEEE Virtual Reality Annual International Symposium, 1993.* IEEE, Sep. 1993, pp. 202–208. 21, 22

[58] J. Colgate, M. Stanley, and J. Brown, "Issues in the haptic display of tool use," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, vol. 3. Los Alamitos, CA, USA: IEEE Computer Society, 1995, p. 3140. 22, 27, 55, 69, 170

[59] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer, "Improving contact realism through event-based haptic feedback," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 219–230, 2006. 22, 23, 29, 83, 146

[60] S. E. Salcudean and T. D. Vlaar, "On the emulation of stiff walls and static friction with a magnetically levitated Input/Output device," *Journal of Dynamic Systems, Measurement, and Control*, vol. 119, no. 1, pp. 127–132, 1997. 23, 28

[61] A. M. Okamura, M. R. Cutkosky, and J. T. Dennerlein, "Reality-based models for vibration feedback in virtual environments," *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 245–252, 2001. 23, 29

[62] J. Sreng, "Contribution to the study of visual, auditory and haptic rendering of information of contact in virtual environments," Ph.D. dissertation, Institut National des Sciences Appliquées, 2008. 23

[63] A. Okamura, "Literature survey of haptic rendering, collision detection, and object modeling," Stanford University, Dextrous Manipulation Lab, Technical report, 1999. 24

[64] S. D. Laycock and A. M. Day, "A survey of haptic rendering techniques," *Computer Graphics Forum*, vol. 26, no. 1, pp. 50–65, Mar. 2007. 25, 30

[65] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha, "Six degree-of-freedom haptic display of polygonal models," in *Visualization 2000 Proceedings.* IEEE, 2000, pp. 139–146. 25

[66] M. C. Lin and J. F. Canny, "A fast algorithm for incremental distance calculation," in *IEEE International Conference on Robotics and Automation.* IEEE, 1991, pp. 1008–1014 vol.2. 25

[67] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," in *Proceedings of SIGGRAPH.* ACM Press, 1988, pp. 289–298. 25

[68] D. E. Johnson and P. Willemsen, "Six Degree-of-Freedom haptic rendering of complex polygonal models," in *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2003. 25

[69] D. E. Johnson and E. Cohen, "A framework for efficient minimum distance computations," in *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, vol. 4. IEEE, 1998, pp. 3678–3684 vol.4. 25

[70] Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha, "Six-Degree-of-Freedom haptic display using localized contact computations," in *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, ser. HAPTICS '02. Washington, DC, USA: IEEE Computer Society, 2002, p. 209–. 25

[71] M. Ortega, S. Redon, and S. Coquillart, "A six Degree-of-Freedom God-Object method for haptic display of rigid bodies with surface properties," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 458–469, 2007. 26

[72] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, vol. 3. Los Alamitos, CA, USA: IEEE Computer Society, 1995, p. 3146. 26, 33

[73] S. Redon, A. Kheddar, and S. Coquillart, "Fast continuous collision detection between rigid bodies," *Computer Graphics Forum*, 2002, the definitive version is available at www.blackwell-synergy.com. 26

[74] M. A. Otaduy and M. C. Lin, "Sensation preserving simplification for haptic rendering," in *ACM SIGGRAPH 2003 Papers.* San Diego, California: ACM, 2003, pp. 543–553. 26

[75] M. Otaduy and M. Lin, "A modular haptic rendering algorithm for stable and transparent 6-DOF manipulation," *Robotics, IEEE Transactions on*, vol. 22, no. 4, pp. 751–762, 2006. 26

[76] D. E. Johnson, P. Willemsen, and E. Cohen, "Six Degree-of-Freedom haptic rendering using spatialized normal cone search," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, pp. 661–670, 2005. 26

[77] M. Minsky, O.-y. Ming, O. Steele, F. P. Brooks, and M. Behensky, "Feeling and seeing: issues in force display," in *I3D '90 Proceedings of the 1990 symposium on Interactive 3D graphics.* ACM Press, 1990, pp. 235–241. 27

[78] J. Siira and D. K. Pai, "Haptic texturing-a stochastic approach," in *, 1996 IEEE International Conference on Robotics and Automation, 1996. Proceedings*, vol. 1. IEEE, 1996, pp. 557–562 vol.1. 27

[79] M. A. Otaduy, N. Jain, A. Sud, and M. C. Lin, "Haptic display of interaction between textured models," in *IEEE Visualization, 2004*. IEEE, 2004, pp. 297– 304. 27

[80] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 401–408. 27, 28, 41

[81] M. Renz, C. Preusche, M. Pötke, H.-p. Kriegel, and G. Hirzinger, "Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm," in *Eurohaptics*, 2001, pp. 149—154. 27

[82] M. Wan and W. A. McNeely, "Quasi-Static approximation for 6 Degrees-of-Freedom haptic rendering," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. IEEE Computer Society, 2003, p. 34. 28

[83] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Voxel-Based 6DOF haptic rendering improvements," in *Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE Computer Society, 2006. 28

[84] E. Ruffaldi, D. Morris, F. Barbagli, K. Salisbury, and M. Bergamasco, "Voxel-Based haptic rendering using implicit sphere trees," in *Proceedings of the 2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Washington, DC, USA: IEEE Computer Society, 2008, p. 319–325. 28

[85] A. M. Okamura, J. T. Dennerlein, and R. D. Howe, "Vibration feedback models for virtual environments," in *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, vol. 1. IEEE, 1998, pp. 674–679 vol.1. 29

[86] J. P. Fiene and K. J. Kuchenbecker, "Shaping Event-Based haptic transients via an improved understanding of real contact dynamics," in *EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint*. IEEE, 2007, pp. 170–175. 29

[87] J. Sreng, A. Lecuyer, C. Andriot, and B. Arnaldi, "Spatialized haptic rendering: Providing impact position information in 6DOF haptic simulations using vibrations," in *IEEE Virtual Reality Conference*. IEEE, 2009, pp. 3–9. 29

[88] P. Jacobs, M. J. Fu, and M. C. Cavusoglu, "High fidelity haptic rendering of frictional contact with deformable objects in virtual environments using multi-rate simulation," *The International Journal of Robotics Research*, vol. 29, pp. 1778–1792, Sep. 2010. 30

[89] N. Swarup, "Haptic interaction with deformable objects using real-time dynamic simulation," Thesis, Massachusetts Institute of Technology, 1995, thesis (M.S.)–Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 1995. 31

[90] J. Corso, J. Chhugani, and A. Okamura, "Interactive haptic rendering of deformable surfaces based on the medial axis transform," in *Eurohaptics*, 2002, pp. 92–98. 31

[91] D. Weiss and A. Okamura, “Haptic rendering of tissue cutting with scissors,” *Medicine Meets Virtual Reality*, vol. 12, pp. 407–409, 2004. 31

[92] M. C. Cavusoglu and F. Tendick, “Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments,” in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA ’00*, vol. 3. IEEE, 2000, pp. 2458–2465 vol.3. 31

[93] D. Daulignac, M. C. Cavusoglu, and C. Laugier, “Modeling the dynamics of the human thigh for a realistic echographic simulator with force feedback,” in *Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI ’99. London, UK: Springer-Verlag, 1999, p. 1191–1198. 31

[94] M. Bro-Nielsen and S. Cotin, “Real-time volumetric deformable models for surgery simulation using finite elements and condensation,” *Computer Graphics Forum*, vol. 15, no. 3, pp. 57–66, 1996. 32

[95] D. C. Popescu and M. Compton, “A model for efficient and accurate interaction with elastic objects in haptic virtual environments,” in *GRAPHITE ’03 Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia.* ACM Press, 2003, p. 245. 32

[96] M. Sedef, E. Samur, and C. Basdogan, “Real-Time Finite-Element simulation of linear viscoelastic tissue behavior based on experimental data,” *IEEE Computer Graphics and Applications*, vol. 26, no. 6, pp. 58–68, 2006. 32

[97] I. Peterlík, M. Sedef, C. Basdogan, and L. Matyska, “Real-time visio-haptic interaction with static soft tissue models having geometric and material nonlinearity,” *Computers & Graphics*, vol. 34, no. 1, pp. 43–54, 2010. 32

[98] D. L. James and D. K. Pai, “A unified treatment of elastostatic contact simulation for real time haptics,” in *ACM SIGGRAPH 2005 Courses.* Los Angeles, California: ACM, 2005, p. 141. 32

[99] J. Barbič and D. L. James, “Six-DoF haptic rendering of contact between geometrically complex reduced deformable models,” *IEEE Transactions on Haptics*, vol. 1, no. 1, pp. 39–52, 2008. 32, 33

[100] I. Peterlík, M. Nouicer, C. Duriez, S. Cotin, and A. Kheddar, “Constraint-Based haptic rendering of multirate compliant mechanisms,” *IEEE Transactions on Haptics*, vol. 4, no. 3, pp. 175–187, 2011. 32

[101] G. Saupin, C. Duriez, and S. Cotin, “Contact model for haptic medical simulations,” in *Proceedings of the 4th international symposium on Biomedical Simulation.* London, UK: Springer-Verlag, 2008, pp. 157–165. 33, 34

[102] M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, “Stable real-time deformations,” in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation.* San Antonio, Texas: ACM, 2002, pp. 49–54. 33, 156

[103] S. De, J. Kim, and M. A. Srinivasan, “A meshless numerical technique for physically based real time medical simulations,” in *Medicine Meets Virtual Reality*, 2001, pp. 113–118. 34, 35

[104] Y. Visell, B. L. Giordano, G. Millet, and J. R. Cooperstock, “Vibration influences haptic perception of surface compliance during walking,” *PLoS ONE*, vol. 6, no. 3, p. e17697, 2011. 35, 146

[105] A. W. Law, B. V. Peck, Y. Visell, P. G. Kry, and J. R. Cooperstock, “A multi-modal floor-space for experiencing material deformation underfoot in virtual reality,” in *IEEE International Workshop on Haptic Audio visual Environments and Games, 2008. HAVE 2008.* IEEE, 2008, pp. 126–131. 35, 36

[106] M. Muller, J. Stam, D. James, and N. Thürey, “Real time physics class notes,” Los Angeles, California, United States, 2008. 36, 151, 156, 161

[107] J. Mora and W. Lee, “Real-Time 3D fluid interaction with a haptic user interface,” in *3D User Interfaces*, vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2008, pp. 75–81. 37, 62

[108] M. Miyaura, T. Narumi, K. Nishimura, T. Tanikawa, and M. Hirose, “Olfactory feedback system to improve the concentration level based on biological information,” in *IEEE Virtual Reality Conference (VR).* IEEE, 2011, pp. 139–142. 38, 148

[109] T. Narumi, S. Nishizaka, T. Kajinami, T. Tanikawa, and M. Hirose, “Augmented reality flavors.” ACM Press, 2011, p. 93. 38, 148

[110] P. Richard, G. Burdea, D. Gomez, and P. Coiffet, “A comparison of haptic visual and auditive force feedback for deformable virtual objects,” in *Proceedings of International Conference on Artificial Reality and Tele-existence*, 1994. 38

[111] P. Richard, G. Birebent, P. Coiffet, G. C. Burdea, D. Gomez, and N. A. Langrana, “Effect of frame rate and force feedback on virtual object manipulation,” *Presence*, pp. 95–108, 1996. 38, 80

[112] B. Petzold, M. F. Zaeh, B. Faerber, B. Deml, H. Egermeier, J. Schilp, and S. Clarke, “A study on visual, auditory, and haptic feedback for assembly tasks,” *Presence: Teleoperators and Virtual Environments*, vol. 13, pp. 16–21, 2004. 38, 80

[113] Y. Zhang, T. Fernando, H. Xiao, and A. R. L. Travis, “Evaluation of auditory and visual feedback on task performance in a virtual assembly environment,” *Presence: Teleoperators and Virtual Environments*, vol. 15, pp. 613–626, 2006. 38, 80, 141

[114] A. Kontogeorgakopoulos and C. Cadoz, “Designing and synthesizing Delay-Based digital audio effects using the CORDIS ANIMA physical modeling formalism,” in *Sound and Music Computing Conference*, 2008. 40

[115] ACROE, “ACROE media library,” http://acroe.imag.fr/mediatheque, 2011. 40

[116] C. Cadoz, A. Luciani, J. Florens, C. Roads, and F. Chadabe, “Responsive input devices and sound synthesis by stimulation of instrumental mechanisms: The cordis system,” *Computer Music Journal*, vol. 8, no. 3, pp. 60–73, 1984, ArticleType: research-article / Full publication date: Autumn, 1984 / Copyright © 1984 The MIT Press. 40

[117] M. Rath, F. Avanzini, N. Bernardini, G. Borin, F. Fontana, L. Ottaviani, and D. Rocchesso, “An introductory catalog of computer-synthesized contact sounds, in real-time,” in *Proc. of the XIV Colloquium on Musical Informatics*, Firenze, Italy, 2003, pp. 103–108. 40

[118] R. Gupta, D. Whitney, and D. Zeltzer, “Prototyping and design for assembly analysis using multimodal virtual environments,” *Computer-Aided Design*, vol. 29, no. 8, pp. 585–597, 1997. 40

[119] H. Baier, M. Buss, F. Freyberger, J. Hoogen, P. Kammermeier, and G. Schmidt, “Distributed PC-based haptic, visual and acoustic telepresence system-experiments in virtual and remote environments,” in *IEEE Virtual Reality, 1999. Proceedings.* IEEE, 1999, pp. 118–125. 41

[120] W. Wan, S. Gao, Q. Peng, G. Dai, and F. Zhang, “MIVAS: a multi-modal immersive virtual assembly system,” in *Proceedings of ASME Design Engineering Technical Conferences*, 2004. 41

[121] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, “The CAVE: audio visual experience automatic virtual environment,” *Communications of the ACM*, vol. 35, pp. 64–72, Jun. 1992. 41, 95

[122] I. n. Diaz, J. Hernantes, I. Mansa, A. Lozano, D. Borro, J. J. Gil, and E. Sánchez, “Influence of multisensory feedback on haptic accessibility tasks,” *Virtual Reality*, vol. 10, pp. 31–40, Apr. 2006. 41

[123] K. v. d. Doel and D. K. Pai, “The sounds of physical shapes,” *Presence: Teleoper. Virtual Environ.*, vol. 7, no. 4, pp. 382–395, 1998. 41

[124] K. v. d. Doel, P. G. Kry, and D. K. Pai, “FoleyAutomatic: physically-based sound effects for interactive simulation and animation,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques.* ACM, 2001, pp. 537–544. 41

[125] E. Richard, A. Tijou, P. Richard, and J. Ferrier, “Multi-modal virtual environments for education with haptic and olfactory feedback,” *Virtual Reality*, vol. 10, pp. 207–225, Oct. 2006. 42

[126] R. Bridson and M. Muller-Fischer, “Fluid simulation: SIGGRAPH 2007 course,” in *ACM SIGGRAPH 2007 courses.* San Diego, California: ACM, 2007, pp. 1–81. 46, 157

[127] L. B. Lucy, “A numerical approach to the testing of the fission hypothesis,” *The Astronomical Journal*, vol. 82, p. 1013, 1977. 46

[128] R. A. Gingold and J. J. Monaghan, “Smoothed particle hydrodynamics - theory and application to non-spherical stars,” *Royal Astronomical Society, Monthly Notices*, vol. 181, pp. 375–389, 1977. 46

[129] J. Stam and E. Fiume, “Depicting fire and other gaseous phenomena using diffusion processes,” in *SIGGRAPH.* ACM Press, 1995, pp. 129–136. 46

[130] M. Desbrun and M. Cani, “Smoothed particles: a new paradigm for animating highly deformable bodies,” in *Proceedings of the Eurographics workshop on Computer animation and simulation ’96.* Poitiers, France: Springer-Verlag New York, Inc., 1996, pp. 61–76. 46, 47

[131] C. Pozrikidis, *Numerical Computation in Science and Engineering*, illustrated edition ed. Oxford University Press, USA, Apr. 1998. 47, 53, 69

[132] T. Harada, S. Koshizuka, and Y. Kawaguchi, “Smoothed particle hydrodynamics on GPUs,” in *Proc. of Computer Graphics International*, 2007, pp. 70, 63. 49, 62

[133] Y. Zhang, B. Solenthaler, and R. Pajarola, “Adaptive sampling and rendering of fluids on the GPU,” in *Proceedings of Symposium on Point-Based Graphics*, Aug. 2008. 49, 62

[134] NVIDIA, “NVIDIA CUDA programming guide 3.2,” 2008. 49, 73

[135] S. Green, “CUDA particles,” 2008. 49

[136] C. R. Johns and D. A. Brokenshire, “Introduction to the cell broadband engine architecture,” *IBM Journal of Research and Development*, vol. 51, no. 5, pp. 503 –519, 2007. 51

[137] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, R. Espasa, E. Grochowski, T. Juan, and P. Hanrahan, “Larrabee: a many-core x86 architecture for visual computing,” *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–15, 2008. 51

[138] B. Solenthaler, J. Schläfli, and R. Pajarola, “A unified particle model for fluid-solid interactions,” *Comput. Animat. Virtual Worlds*, vol. 18, no. 1, pp. 69–82, 2007. 51, 52, 66, 67, 68, 71, 80, 168, 169, 172

[139] N. Bell, Y. Yu, and P. J. Mucha, “Particle-based simulation of granular materials,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Los Angeles, California: ACM, 2005, pp. 77–86. 52

[140] J. J. Monaghan, “Smoothed particle hydrodynamics,” *Reports on Progress in Physics*, vol. 68, no. 8, pp. 1703–1759, 2005. 52

[141] M. Muller, S. Schirm, M. Teschner, B. Heidelberger, and M. Gross, “Interaction of fluids with deformable solids,” *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 159–171, 2004. 52, 65

[142] R. Keiser, B. Adams, P. Dutre, L. Guibas, and M. Pauly, “Multiresolution Particle-Based fluids,” ETH Zurich, Tech. Rep., 2006. 52

[143] M. Muller, B. Solenthaler, R. Keiser, and M. Gross, “Particle-based fluid-fluid interaction,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Los Angeles, California: ACM, 2005, pp. 237–244. 53, 85, 174

[144] M. Harris, “Optimizing parallel reduction in CUDA,” 2008. 54

[145] I. D. Rosenberg and K. Birdwell, “Real-time particle isosurface extraction,” in *Proceedings of the 2008 symposium on Interactive 3D graphics and games*. Redwood City, California: ACM, 2008, pp. 35–43. 55

[146] M. Muller, S. Schirm, and S. Duthaler, “Screen space meshes,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. San Diego, California: Eurographics Association, 2007, pp. 9–15. 55

[147] H. Cords and O. Staadt, "Instant liquids," in *Poster Proceedings of ACM Siggraph/Eurographics Symposium on Computer Animation.* Dublin, Ireland: Eurographics Association, 2008. 55

[148] W. J. v. d. Laan, S. Green, and M. Sainz, "Screen space fluid rendering with curvature flow," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games.* Boston, Massachusetts: ACM, 2009, pp. 91–98. 55, 56, 58

[149] K. Hegeman, N. Carr, and G. Miller, "Particle-Based fluid simulation on the GPU," in *Lecture Notes on Computer Science*, 2006, pp. 228–235. 62

[150] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, and M. Gross, "A unified lagrangian approach to solid-fluid animation," in *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, 2005, pp. 125 – 148. 65

[151] M. Becker, M. Ihmsen, and M. Teschner, "Corotated SPH for deformable solids," in *Proceedings of the Eurographics Workshop on Natural Phenomena.* Munich, Germany: Eurographics Association, 2009. 66, 67, 80, 168, 172

[152] O. Lazarevych, J. Spillmann, C. Renner, G. Szekely, and M. Harders, "Friction handling for Penalty-Based methods." in *VRIPHYS'09*, 2009, pp. 135–144. 72

[153] E. G. Richardson, "The sounds of impact of a solid on a liquid surface," *Proceedings of the Physical Society. Section B*, vol. 68, no. 8, pp. 541–547, 1955. 84, 86, 87, 174

[154] G. J. Franz, "Splashes as sources of sound in liquids," *The Journal of the Acoustical Society of America*, vol. 31, no. 8, pp. 1080–1096, 1959. 84

[155] K. v. d. Doel, "Physically based models for liquid sounds," *ACM Trans. Appl. Percept.*, vol. 2, no. 4, pp. 534–546, 2005. 84, 85, 87, 174

[156] M. Minnaert, "On musical air-bubbles and the sounds of running water," *Philosophical Magazine Series 7*, vol. 16, no. 104, pp. 235 – 248, 1933. 85, 87, 174

[157] C. Drioli and D. Rocchesso, "Acoustic rendering of Particle-Based simulation of liquids in motion," in *Proceedings of the Int. Conference on digital Audio Effects*, Como, Italy, Sep. 2009. 85, 86, 90

[158] W. Moss, H. Yeh, J. Hong, M. C. Lin, and D. Manocha, "Sounding liquids: Automatic sound synthesis from fluid simulation," *ACM Trans. Graph.*, vol. 29, no. 3, pp. 1–13, 2010. 85, 87, 88, 90, 174

[159] M. Lesser, "Thirty years of liquid impact research: a tutorial review," *Wear*, vol. 186-187, no. Part 1, pp. 28–34, 1995. 86

[160] S. Howison, J. Ockendon, and J. Oliver, "Deep- and shallow-water slamming at small and zero deadrise angles," *Journal of Engineering Mathematics*, vol. 42, no. 3, pp. 373–388, 2002. 86

[161] M. S. Longuet-Higgins, "An analytic model of sound production by raindrops," *Journal of Fluid Mechanics*, vol. 214, pp. 395–410, 1990. 87

[162] N. I. Durlach, A. S. Mavor, and N. R. C. U. C. o. V. R. R. a. Development, *Virtual Reality: Scientific and technological challenges*, 1995. 96

[163] H. Iwata and T. Fujii, "VIRTUAL PERAMBULATOR: a novel interface device for locomotion in virtual environment," in *Proceedings of the 1996 Virtual Reality Annual International Symposium (VRAIS 96)*. IEEE Computer Society, 1996, p. 60. 97, 98

[164] H. Iwata, H. Yano, and H. Tomioka, "Powered shoes," in *ACM SIGGRAPH 2006 Emerging technologies*. Boston, Massachusetts: ACM, 2006, p. 28. 97

[165] H. Iwata, H. Yano, and M. Tomiyoshi, "String walker," in *ACM SIGGRAPH 2007 emerging technologies*. San Diego, California: ACM, 2007, p. 20. 97

[166] H. Iwata and Y. Yoshida, "Path reproduction tests using a torus treadmill," *Presence: Teleoper. Virtual Environ.*, vol. 8, no. 6, pp. 587–597, 1999. 97, 98, 100, 101

[167] H. Schmidt, S. Hesse, R. Bernhardt, and J. Krüger, "HapticWalker—a novel haptic foot device," *ACM Trans. Appl. Percept.*, vol. 2, no. 2, pp. 166–180, 2005. 98

[168] J. M. Hollerbach, Y. Xu, R. R. Christensen, and S. C. Jacobsen, "Design specifications for the second generation sarcos treadport locomotion interface," *Haptic Symposium, Proceedings of ASME Dynamic Systems and Control Division*, vol. 69, pp. 1293—1298, 2000. 99

[169] J. M. Hollerbach, R. Mills, D. Tristano, R. R. Christensen, W. B. Thompson, and Y. Xu, "Torso force feedback realistically simulates slope on Treadmill-Style locomotion interfaces," *The International Journal of Robotics Research*, vol. 20, no. 12, pp. 939–952, 2001. 99

[170] J. M. Hollerbach, D. Checcacci, H. Noma, Y. Yanagida, and N. Tetsutani, "Simulating side slopes on locomotion interfaces using torso forces," in *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*. IEEE Computer Society, 2003, p. 91. 99

[171] H. Noma, "Design for locomotion interface in a large-scale virtual environment. ATLAS: ATR locomotion interface for active Self-Motion," in *Proc. of the ASME Dynamic Systems and control division*, 1998, pp. 111–118. 99, 100

[172] H. Noma, T. Sugihara, and T. Miyasato, "Development of ground surface simulator for Tel-E-Merge system," in *Proceedings of the IEEE Virtual Reality 2000 Conference*. IEEE Computer Society, 2000, p. 217. 99, 100

[173] L. Bouguila, M. Ishii, and M. Sato, "Realizing a new step-in-place locomotion interface for virtual environment with large display system," in *Proceedings of the workshop on Virtual environments 2002*. Barcelona, Spain: Eurographics Association, 2002, pp. 197–207. 99

[174] R. P. Darken, W. R. Cockayne, and D. Carmein, "The omni-directional treadmill: a locomotion device for virtual worlds," in *Proceedings of the 10th annual ACM symposium on User interface software and technology*. Banff, Alberta, Canada: ACM, 1997, pp. 213–221. 99

[175] M. Schwaiger, T. Thümmel, and H. Ulbrich, "Cyberwalk: Implementation of a ball bearing platform for humans," in *Human-Computer Interaction. Interaction Platforms and Techniques*, 2007, pp. 926–935. 100, 101

[176] H. Tomoyuki, K. Masato, K. Shingo, T. Shuntaro, N. Fumikazu, and I. Minoru, "The ball array treadmill," *Proceedings of the Virtual Reality Society of Japan Annual Conference*, vol. 6th, pp. 211–212, 2001. 100

[177] H. Iwata, H. Yano, H. Fukushima, and H. Noma, "CirculaFloor: a locomotion interface using circulation of movable tiles," in *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality.* IEEE Computer Society, 2005, pp. 223–230. 101

[178] K. J. Fernandes, V. Raja, and J. Eyre, "Cybersphere: the fully immersive spherical projection system," *Commun. ACM*, vol. 46, no. 9, pp. 141–146, 2003. 101, 102

[179] N. N. Latypov, "The virtusphere," 2006. 101, 102

[180] R. Pausch, T. Burnette, D. Brockway, and M. E. Weiblen, "Navigation and locomotion in virtual worlds via flight into hand-held miniatures," in *Proceedings of ACM SIGGRAPH.* ACM, 1995, pp. 399–400. 102, 103

[181] J. N. Templeman, P. S. Denbrook, and L. E. Sibert, "Virtual locomotion: Walking in place through virtual environments," *Presence: Teleoper. Virtual Environ.*, vol. 8, no. 6, pp. 598–617, 1999. 103, 128

[182] V. Interrante, B. Ries, and L. Anderson, "Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments," in *Proceedings of the IEEE Symposium on 3D User Interfaces*, 2007, pp. 167—170. 104

[183] F. Steinicke, G. Bruder, T. Ropinski, and K. H. Hinrichs, "Moving towards generally applicable redirected walking," in *Proceedings of the Virtual Reality International Conference (VRIC).* IEEE Press, 2008, pp. 15—24. 105

[184] T. Peck, M. Whitton, and H. Fuchs, "Evaluation of reorientation techniques for walking in large virtual environments," in *Virtual Reality Conference, 2008. VR '08. IEEE*, 2008, pp. 121–127. 106

[185] S. Razzaque, D. Swapp, M. Slater, M. C. Whitton, and A. Steed, "Redirected walking in place," in *Proceedings of the workshop on Virtual environments 2002.* Barcelona, Spain: Eurographics Association, 2002, pp. 123–130. 106

[186] L. Kohli, E. Burns, D. Miller, and H. Fuchs, "Combining passive haptics with redirected walking," in *Proceedings of the 2005 international conference on Augmented tele-existence.* Christchurch, New Zealand: ACM, 2005, pp. 253–254. 106, 107, 108

[187] G. Bruder, F. Steinicke, and K. H. Hinrichs, "Arch-Explore: a natural user interface for immersive architectural walkthroughs," in *IEEE Symposium on 3D User Interfaces, 2009. 3DUI 2009.* IEEE, 2009, pp. 75–82. 106, 129

[188] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, "Motion compression for telepresent walking in large target environments," *Presence: Teleoper. Virtual Environ.*, vol. 13, no. 1, pp. 44–60, 2004. 107

[189] J. Su, "Motion compression for telepresence locomotion," *Presence: Teleoper. Virtual Environ.*, vol. 16, no. 4, pp. 385–398, 2007. 108

[190] D. Engel, C. Curio, L. Tcheang, B. Mohler, and H. H. Bülthoff, "A psychophysically calibrated controller for navigating through large environments in a limited free-walking space," in *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*. Bordeaux, France: ACM, 2008, pp. 157–164. 107, 108

[191] E. A. Suma, D. M. Krum, and M. Bolas, "Redirection on mixed reality walking surfaces," in *IEEE VR Workshop on Perceptual Illusions in Virtual Environments*, 2011, p. 33–35. 108, 109

[192] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte, "Leveraging change blindness for redirection in virtual environments," in *2011 IEEE Virtual Reality Conference (VR)*. IEEE, 2011, pp. 159–166. 108

[193] L. Dominjon, A. Lecuyer, J. Burkhardt, G. Andrade-Barroso, and S. Richir, "The "Bubble" technique: interacting with large virtual environments using haptic devices with limited workspace," in *Proceedings of World Haptics conference*, 2005, pp. 639–640. 112, 114, 176

[194] L. Dominjon, A. Lécuyer, J. Burkhardt, and S. Richir, "A comparison of three techniques to interact in large virtual environments using haptic devices with limited workspace," *Advances in Computer Graphics*, vol. 4035, pp. 288–299, 2006. 141

[195] M. Becker, H. Tessendorf, and M. Teschner, "Direct forcing for lagrangian Rigid-Fluid coupling," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 493–503, 2009. 145

[196] A. Lecuyer, J. Burkhardt, S. Coquillart, and P. Coiffet, ""Boundary of illusion": an experiment of sensory integration with a pseudo-haptic system," in *Virtual Reality, 2001. Proceedings. IEEE*, 2001, pp. 115–122. 146

[197] D. Baraff, "Rigid body simulation," Los Angeles, California, United States, 2001. 151

[198] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. ACM, 1987, pp. 205–214. 153

[199] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," *Computer Graphics Forum*, vol. 25, no. 4, pp. 1–24, 2006. 153

[200] R. Bridson, *Fluid Simulation for Computer Graphics*. A K Peters, Sep. 2008. 157

[201] F. H. Harlow and J. E. Welch, "Numerical calculation of Time-Dependent viscous incompressible flow of fluid with free surface," *Physics of Fluids*, vol. 8, p. 2182, 1965. 158

Abstract

Virtual Reality allows the simulation and interaction with Virtual Environments (VE) through different sensory modalities. However, interacting with complex physically based VE, such as non-rigid or large environments, presents many challenges in terms of interaction and sensory feedback.

The first part of this Ph.D. thesis addresses haptic and multimodal feedback issues during manipulation of non-rigid media. We first present a novel approach for 6 degrees of freedom haptic interaction with fluids, allowing the generation of force feedback from viscous fluids through arbitrary-shaped rigid bodies. This approach is extended by including the haptic interaction with deformable bodies, thus allowing a unified haptic interaction with the different states of matter. A perceptual experiment showed that users could efficiently identify the different states through the haptic modality alone. Then, we introduce a novel vibrotactile fluid rendering model, leveraging previous knowledge on fluid sound synthesis. Through this approach, we allow the interaction with fluids with multimodal feedback, through vibrotactile, kinesthetic, acoustic and visual sensory channels.

The second part of this Ph.D. thesis addresses interaction issues during walking navigation of large VE. Since the VE is often larger than the available real workspace, we introduce a novel navigation metaphor that informs users about the real physical boundaries. Using hybrid position/rate control, this technique provides a simple and intuitive metaphor for a navigation safe from collisions and breaks of immersion. Other workspaces, such as CAVE-like environments, present rotation boundaries due to missing screens. Thus, we present three novel metaphors dealing with these additional boundaries. Overall, the evaluation of these navigation techniques showed that they efficiently fulfilled their objectives while being highly appreciated by users.

Résumé

La Réalité Virtuelle permet de simuler et d'interagir avec des Environnements Virtuels (EV) à travers différentes modalités sensorielles. Cependant, l'interaction avec des EV basés physique et complexes, comme des environnements non rigides ou larges, présente plusieurs défis en termes d'interaction et de retours sensoriels.

Dans la première partie de cette thèse, nous abordons la manipulation de milieux non rigides avec du retour haptique et multimodal. Nous présentons tout d'abord une nouvelle approche pour l'interaction haptique à 6 degrés de liberté avec des fluides. Cette approche permet la génération de retours de force lors de l'interaction avec des fluides visqueux par le biais d'objets rigides de forme arbitraire. Nous étendons ensuite cette approche pour inclure l'interaction haptique avec des objets déformables, menant donc à une interaction haptique unifiée avec les différents états de la matière. Une expérience perceptuelle nous a permis de montrer que les utilisateurs peuvent identifier de façon efficace les différents états en n'utilisant que la modalité haptique. Ensuite, nous présentons un nouveau modèle vibrotactile pour le rendu de fluides, tirant parti de connaissances dans la synthèse de son de fluides. A travers cette approche, nous rendons possible l'interaction avec des fluides tout en générant des retours multimodaux, utilisant les canaux sensoriels vibrotactile, kinesthésique, acoustique et visuel.

Dans la seconde partie de cette thèse, nous abordons la navigation basée sur la marche dans des EV larges. Comme les EV sont souvent plus larges que l'espace de travail réel, nous présentons une nouvelle technique de navigation qui permet à l'utilisateur de connaître de façon immersive les limites de son espace de travail en translation. En utilisant un contrôle hybride en position/vitesse, cette technique fournit une métaphore simple et intuitive pour une navigation libre de collisions et de ruptures d'immersion. Comme certains espaces de travail présentent aussi des limites en rotation dues à des écrans manquants, comme dans le cas d'un CAVE, nous proposons ensuite trois nouvelles métaphores de navigation abordant ce problème supplémentaire. L'évaluation de ces techniques de navigation a permis de montrer qu'elles remplissent efficacement leurs objectifs tout en étant très appréciées par les utilisateurs.