UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DAVID STEEVEN VILLA SALAZAR

Touching is believing: Exploring Physics-based simulation and haptics to feel virtual worlds

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Anderson Maciel

Porto Alegre
September 2019

CIP — CATALOGING-IN-PUBLICATION

Villa Salazar, David Steeven

Touching is believing: Exploring Physics-based simulation and haptics to feel virtual worlds / David Steeven Villa Salazar. – Porto Alegre: PPGC da UFRGS, 2019.

122 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2019. Advisor: Anderson Maciel.

1. Physics-based simulation. 2. Haptics. 3. Virtual Reality. 4. Robotics. I. Maciel, Anderson. II. Título.

“La mano toca e investiga; el resto del cuerpo siente, recibe. La mano es activa, se prolonga para tocar, se adelanta a recorrer los objetos, palpándolos, acariciándolos con esos dedos cuya envoltura es toda sensibilidad. . . ”

— FERNANDO GONZALEZ OCHOA

AKNOWLEDGEMENTS

The path to the knowledge is long and full of wonderful people, so I would like to start acknowledging the professors who led me to this point. First of all, to Professor Dr. Anderson Maciel, who allowed me to work in his lab since 2017. Probably I wouldn't find a better place nor a better guide to starting my scientific career and face all the scientific issues that an early-scientific faces in this stage. Second, to Professor Luciana Nedel, who although is not my official co-advisor, behaved like that during all this way long. Also Professor Dr. Rafael Torchelsen was a tremendous support during the conception of the graphic part of this work. And also thanks to all the professors from the institute from informatics who provided me knowledge: **The most precious good**

A special thanks to Dr. Claudio Pacchierotti and the RAINBOW Team from INRIA Rennes, France for the support, the collaboration, and the trustworthiness that put on me. I feel fortunate to have had the chance of working with such a big research team. In this matter, I would also like to thanks Professor Dr. Maud Marchal From HYBRID team, who introduced me to user testing. Also from HYBRYD team, I want to thank Victor Mercado and Gerard Gallagher for the sincere friendship, Cheers!.

I definitively have to thank Laura Amaya, for her friendship during all the time, thanks for the advises good wishes and support in general. Also, the friends I made in Brazil were a big motivation to improve, to grow, to enjoy. Among them, my friend Aasim Kurshid deserves a special thank you. Thanks to Victor Oliveira for being a real friend but also a great mentor. You guys made me feel at home.

Special thanks to my coauthor Abel Ticona for his support and his hard work, the results related to Physics-based animation wouldn't be possible without his help.

To Jaime Riascos, who motivated me to go outside my home country to pursue my dream of being a world-class researcher and then helped me to come to Brazil.

To my family, who is my motivation, who always understand me, even if I have to live far from them.

Gracias Mama, Gracias Papa, Gracias Hermana, Gracias Sobrina, Gracias Tio. Esto es por ustedes.

"You've always been there for me
And so I'll be there for you, you"

[0] This study was funded by Petrobras (Project Annelida), the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and partially by the EU's H2020 research and innovation programme (grant agreement No 801413, project "H-Reality").

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

PBD	Position-based Dynamics
SPH	Smoothed-Particle Hydrodynamics
PBF	Position-based Fluids
XSPH	Extended Position-based Dynamics
LBM	Lattice Boltzmann Method
FLIP	Fluid Implicit Particles
PIC	Particle In Cell
MSD	Mass Springs systems
OSC	Open Sound Protocol Commmunication
VR	Virtual Reality

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Modeling the real is a responsibility which different fields assumed through history, from philosophy to physics. Although the objective is similar, the strategies used to model real-world are different from field to field. Nowadays, the modeling of the world took a more tangible significance: Being in the modeled world, existing in the artificial world instead of the actual world. However, the path to get an artificial world equal to the real world is long and arduous. In this manuscript, we address this issue by exploring methods in computer graphics (Physics-based animation) and haptics to project a sensory representation of the actual world into the artificial world. We report results on Position based dynamics for simulating phase-change phenomena and interaction in VR with physical objects. Also, Ultrasound phased arrays, as well as wearable haptics for stiffness and softness rendering are studied.

Keywords: Physics-based simulation. Haptics. Virtual Reality. Robotics.

Tocar para Crer: Explorando simulação baseada em Física e haptica para sentir mundos virtuais

RESUMO

Modelar o mundo real é uma responsabilidade que diferentes campos já assumiram através da história, da filosofia à física. Embora o objetivo seja semelhante, os meios utilizados para modelar o mundo real são diferentes de campo para campo. Hoje em dia, a modelagem do mundo tomou um significado mais tangível: Estar no mundo modelado, existindo no mundo artificial em vez do mundo real. No entanto, o caminho para obter um mundo artificial igual ao mundo real é longo e árduo. Neste manuscrito, abordamos esta questão explorando métodos em computação gráfica (animação baseada em física) e háptica para projetar uma representação sensorial do mundo real no mundo artificial. Nós relatamos resultados em Position-based Dynamics para simular fenômenos de mudança de fase e interação em RV com objetos físicos. Além disso, são estudadas matrizes de ultrassom faseadas assim como Wearable Haptics para rigidez e suavidade de renderização.

Palavras-chave: animação baseada em física, Haptica, Realidade Virtual, Robotica.

1 INTRODUCTION

From Natural Reality to Artificial Reality

In the 80s, virtual reality emerged with the development of the first commercial Head-Mounted Displays (HMDs), bringing to discussion several aspects of this new parallel (virtual) universe. However, if we look deeper into the previous manifestations of popular art, this search for an alternative existence was present before. As an example, let's take science fiction as the reflection of this social willingness: In 1950 (Fifteen years before Ivan Sutherland develops the first HMD) Ray Bradbury wrote *The Veldt*, a short story describing a virtual nursery which can project any environment in the mind of the visitor. Since that, the discussion around virtuality has increasingly appeared in other fields such as movies, novels, series, books.

In the latter decades, contrasting ordinary reality and VR became more common in the scientific community and society in general. However, somehow, the interpretation of virtual reality has been used arbitrarily even among computer science and interaction researchers; sometimes described as a framework to fake human senses, or the digital representation of reality. There is a vast group of interpretations of virtual reality in the literature, from the metaphysic perspective to the materialists point of view. Consequently, it is imperative to state how the author understands virtual reality. Therefore, it is essential to discriminate VR as the experienced phenomena and VR as the medium used to get into this reality.

Regarding the first, the author would like to state some concepts: we see that virtual reality exists as a broader and independent reality, not as digitalization of natural reality. VR could, in theory, comprehend the ordinary reality, but it does not happen in the opposite way, unconstrained phenomena which are possible in VR not always can be replicated in actual reality. Also, immersion[1] in VR worlds could be achieved by enforcing consistency in the overall environment, as in the real world. However, as our previous experience was obtained in natural reality, our expectations of VR are biased by them. In

[1] "Whether or not a system can be classified as immersive depends crucially on the hardware, software, and peripherals (displays and body sensors) of that system. We use “immersion” as a description of a technology, rather than as a psychological characterization of what the system supplies to the human participant" (SLATER; USOH; STEED, 1995)

this line of reasoning, the explorations presented in this manuscript do not want to mimic the real world but to get a projection of natural reality in VR. In other words: to get similar behaviors using different paths.

We share the interpretation of Myeun-Sook Yoh (YOH, 2001), who states that virtual reality can overlap and, moreover expand ordinary reality. So, under this conceptual framework, not the replication but the projection of *actuality* is the target when it comes to real world-like virtualities, because the nature of the environments is different: i.e., gravity is naturally present in our daily life in ordinary reality, but in the digital representation of the same reality, gravity must be enforced. This interpretation of the virtual worlds can be used to support the use of alternative physics frameworks to describe natural-like behaviors, that is the case of position-based dynamics as an alternative to impulse-based dynamics or Position-based fluids instead of solving the Navier Stokes equations.

Finally, Virtual reality as the meaning for the medium/tools used to get immersed in a simulated world is more straightforward to define. Authors like Jonathan Steuer or Mel Slater made significant contributions to the understanding of this paradigm. Both of them agree on the relevance of looking at this problem by separating the perception of the individual experiencing the virtuality, from the set of tools and devices which enables individuals to experience the virtual world; The keywords of these concepts are *Presence* and *Immersion*. Steuer (STEUER, 1992) and then, Slater (SLATER; USOH; STEED, 1995) defined *Presence* as a function of the coherence of the factors involved; Internal representation of the world [2] and proprioception versus sensory data. However, the internal representation of the world changes from individual to individual, so the same virtual environment could lead to a different degree of presence depending on who experiences it. Therefore, the optimization of the sense of *Presence* could be subjective, while the optimization of *Immersion* can be objectively achieved by improving the quality and richness of the pieces of hardware and software responsible of enabling the individual to be *Present*. Also, Slater defines *Immersion* as a computational system capable of transporting one's to the virtual environment provided of objects, actors, geometry, and dynamics. In the same manners, he remarks the importance of multi-sensory input as well as the egocentric point of view, as it is the way we experience the real world as human/animal beings. In short, *Immersion* refers to technology rather than a psychological response to

[2] As stated above, the internal representation of the real world (Previous experiences) significantly influences in the expectations of the virtual world

the data provided by the technology. In this aspect, we accept a further definition made by Michael Heim in 1998 as follows (MICHAEL, 1998):

> “Virtual reality is a technology that convinces the participant that he or she is actually in another place by substituting the primary sensory input with data received produced by a computer”

This description puts VR as the set of technologies responsible for enabling users to perceive digital sensory experiences (to be immersed) in the same way like interacting with the physical world. In other words, a sensorial supply of missing features when interacting with virtual environments. This should not be confused with sensory substitution, which means switching a sense with another i.e., collision stimulation by using vibrotactile actuators. This final description was previously called Artificial reality instead of Virtual reality for substituting natural reality by (YOH, 2001). Under this definition, there are no chances of subjective interpretation as it proposes VR as a simulated experience, not as a parallel existence. Even if the above definition puts a significant emphasis on the hardware/software component of VR, it is the most accurate reading of the explorations presented in this manuscript. Looking to meet the features of an Artificial world we worked on several fronts (Namely Physics-based Animation, Haptic interaction, and Interaction) to generate a simulated reality.

Objective

- To explore available methods to project a sensory representation of the actual world into the artificial world.

Contributions

This document is composed of four individual works, two on Physics-based simulation and two on haptics. The first chapter groups the works related to computer graphics, and the second chapter groups the works related to haptics, the summary of them is:

Chapter One: Staring at the world

- A full Lagrangian method to simulate bidirectional phase-shifting materials as a function of temperature.

- A latent heat model to drive the transitions between the different states
- A particle-based system to sketch and simulate virtual objects with physical behavior in a VR environment
- User test with professional artists
- Expressivity analysis

Chapter Two: Touching the (artificial) world

- A framework to extend mid-air haptics workspace by adding more rotational dimensions
- We studied a novel approach to enhance softness perception using a wearable tactile device
- We explored active roughness augmentation using the same device
- We conducted two user-studies to evaluate the performance of our wearable device in both situations: Softness and Roughness

Structure of the document

Sherman and Craig (SHERMAN; CRAIG, 2018) argue that there are four essential elements in VR: **A virtual world (VW)**, **Immersion (IM)**, **Sensory feedback (SF)**, **Interactivity (IN)**. Putting the content of this work on the same terms, we have the following structure:

The first chapter focuses mainly on the simulation of natural world phenomena, namely the states of the matter and phase-shifting **(VW)**. Also, the same chapter an interactive simulation within a virtual environment **(VW+IM+IN)** is shown. Through all the chapter, we use alternative models to recreate these natural phenomena. Namely, we gave priority to the internal coherence of the model; to this end, we used only particle-based methods.

The second part of this manuscript focuses on Haptic feedback **(SF)**, but primarily we tried to face the problem of constrained workspace when using haptic feedback **(SF+IN)**, this is why none of the works presented on this sections uses grounded feedback. Here we present a strategy to expand the workspace of ultrasound-based mid-air haptics. Moreover, we study softness augmentation by using a wearable device **(SF+IN)**.

2 STARING AT THE WORLD

Several factors directly influence on how successful the user experience is, such as the quality of the graphics and 3D models. However, this static part by itself does not produce a full VR or AR experience. Elements' behavior, as well as the way they move and how the user interacts with them, also play a fundamental role.

Physically based animation has increased its importance in animation movies, the new generation of games and immersive virtual environments in the later years. Capturing real-world physical behaviors is a continuous objective widely pursued by the computer graphics community. Phase transition phenomena have long represented an exciting challenge to study in the context of animation. The cost to simulate the complex non-linear physical laws involved is prohibitive for interactive animation, in such a way that previous methods focused on off-line simulation (NEALEN et al., 2006). It is necessary to build a heat-based model to govern the phase change phenomena and, furthermore, simplify the thermodynamic model to a temperature-based model to efficiently control the different phase transitions (GAO et al., 2017b). Moreover, to correctly simulate real-world situations, an object should account for the thermodynamic properties of its composing material when transitioning between solid, liquid and gas phases. Classic methods such as Finite Elements were used to computationally model these natural phenomena. Unfortunately, those models entail high computational cost due to the complex calculations involved.

Usually, looking for minimizing the computation-time, designers choose to not use physically-accurate objetcs. Therefore, adopting kinematic behaviors is quite popular. However, for a true VR experience, one has to consider physical behaviors of objects made of different materials, as well as simulate the ways people interact with them.

In the first part of this chapter, we propose a full Lagrangian model to simulate phase transitions between solid, liquid and gas states. Our model is sufficiently general to allow a continuous representation of all the basic first-order phase transitions. We based our model on the Position-based dynamics (PBD) framework (MÜLLER et al., 2007) due to its proven stability and its capacity to simulate physical phenomena with significantly lower computational cost than methods based on differential equations. Using this framework, and inheriting its functionality and robustness, we can handily couple all the states modifying the existing constraints that model solids and fluids in order to obtain phase-shifting effects. We demonstrate that only by using PBD and the methods previously demonstrated within that framework (for instance, SPH to calculate density constraint), it is possible to achieve plausible results when replicating thermodynamic phenomena.

As complement, in the second part of this chapter, with the objective of providing a way to interact with the virtual world, we propose a novel immersive sketching application to create objects with different materials. Our solution allows physically-based interactions between objects in real-time using position-based dynamics (PBD). This PBD-based model is also unconditionally stable since we optimize position constraints instead of integrating the next positions of the bodies. It is possible to create expressive dynamic-sketches involving several types of physical behaviors such as rigid solid bodies, liquids, gases, soft solids, and clothes. As an example, it is possible to change the flow of a river by adding or digging into the ground while the river is flowing, or even to create beautiful waterfalls using the same methods. The application also allows the user to see in real time the interaction of recently created soft objects or even cloths with the existing environment, including other sketched objects.

2.1 Background

PBD is a particle-based animation technique that uses a set of constraints to calculate the positions of the particles in each timestep. It tries to fit the position of each particle inside the space, based on the available constraints on the particles. A particle can have an arbitrary number of constraints, but a constraint must have at least two particles. As the solver must iterate to set the positions of the particles based on the constraint available space and trying to satisfy each constraint, we can see this as an optimization problem. This is a successful way to simulate a wide variety of bodies like clothes, deformable, rods, elastics but is not capable of simulating by itself fluid volumes or rigid bodies. To solve that issue it is common to introduce other techniques as Smoothed-particle hydrodynamic, indeed in 2013 this integration was named Position-based fluids (PBF) (MACKLIN; MÜLLER, 2013) and makes possible to simulate fluids within the same frameworks as PBD. Finally to simulate rigid bodies, the most common way to use shape matching (MACKLIN et al., 2014).

2.1.1 Extended Position Based Dynamics

Originally developed by Müller and Heidelberg in 2007 (MÜLLER et al., 2007), PBD is governed by a set of constraints C that must satisfy an equality (or inequality), i.e. $C_i\left(x+\Delta x\right)=0$, or its linearized representation by a series of Newton steps:

$$C_i\left(x+\Delta x\right)\approx C_i\left(x\right)+\nabla C_i\left(x\right)\Delta x. \tag{2.1}$$

where $x=\left[x_1,x_2,...,x_1\right]^T$ is the vector of positions. Equation 2.1 is centered around x and Δx is restricted to take values along the constraint gradient $\nabla C_i\left(x\right)$ (MACKLIN et al., 2014). In contrast with the regular PBD, in extended PBD (XPBD) (MACKLIN; MÜLLER; CHENTANEZ, 2016), the scaling factor λ used to conserve the linear and angular momenta is a delta. In that way, the positon correction is given by Equation 2.2:

$$\Delta x=\nabla C_i\left(x\right)w_i\Delta\lambda. \tag{2.2}$$

Where w_i represents the inverse masses and $\Delta\lambda$ is given by:

$$\Delta\lambda=-\frac{C_i\left(x\right)-\lambda\tilde{\alpha}}{\sum_j w_j\left|\nabla C_j\left(x\right)\right|^2+\tilde{\alpha}}. \tag{2.3}$$

With $\tilde{\alpha}=\alpha/\Delta t^2$ and α the compliance of the material. λ needs to be stored in each iteration:

$$\lambda_{i+1}=\lambda_i+\Delta\lambda. \tag{2.4}$$

2.1.2 Smoothed Particles Hydrodynamics

SPH can be seen as an interpolation scheme between scattered particles that acts in a given radius around a point centered in **r**:

$$A\left(\mathbf{r}\right)=\sum_j m_j\frac{A_j}{\rho_j}W\left(\mathbf{r}-\mathbf{r}_j,h\right). \tag{2.5}$$

Here m_j is the mass of the neighbor particle *j*, ρ_j represents its density, and *W* is a smoothing kernel with support radius *2h*. So, all the summations are performed over all particles j within the radius of the kernel. With exception of the kernel, all parameters are scalars when evaluated in each particle.

Notice that, for the 3D case, while $W\left(r - r_j, h\right)$ is a scalar field ($\mathbb{R}^3 \to \mathbb{R}$), its gradient $\nabla W\left(r - r_j, h\right)$ is a vector field ($\mathbb{R}^3 \to \mathbb{R}^3$). So, the gradient of the function A is obtained just by differentiating the interpolation kernel (CLEARY; MONAGHAN, 1999). In that way, we could use Equation 2.5 to evaluate density. The following result would be obtained:

$$\rho_i = \sum_j m_j W\left(\mathbf{r}_i - \mathbf{r}_j, h\right). \tag{2.6}$$

This means that particle density relies just on its mass and the influence of its neighborhood. In this paper, we work with two kernels, in the same way as Muller et al. 2003 (MÜLLER; CHARYPAR; GROSS, 2003). Poly6 Kernel is used to compute densities and spiky Kernel to calculate gradients.

Viscosity modeling: To damp out non-physical oscillations, we use the XSPH viscosity approach of Schechter and Bridson (SCHECHTER; BRIDSON, 2012) due to its low cost and easiness to manipulate in contrast with the classic XSPH. In this manner, we damp out the noise when updating velocities:

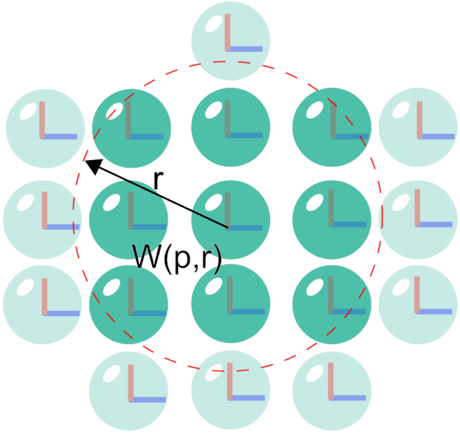
$$\mathbf{v}_i^{next} = \mathbf{v}_i + \mu \sum_j (\mathbf{v}_i - \mathbf{v}_j) W\left(\mathbf{r} - \mathbf{r}_j, h\right). \tag{2.7}$$

The value of μ is tunable, we use this parameter in further sections to generate convection effects.

2.1.3 Position Based Fluids

One of the main issues when working with SPH is the incompressibility, and several works have addressed this problem in the past. Works such as Shao and Edmond (SHAO; LO, 2003), Hu and Adams (HU; ADAMS, 2007) and Solenthaler and Pajarola (SOLENTHALER; PAJAROLA, 2009) notably improved this SPH weakness, but the requirement to have a small time-step to guarantee stability remains. A feasible solution to allow larger time-steps was presented by Muller and Macklin in 2013 (MACKLIN; MÜLLER, 2013). They calculate the fluid particle density as another constraint in

Figure 2.1: Fluid particles: Fluids are governed by a density constraint which depends on a radius and a kernel. This kernel defines the influence of the neighboring particles over the interest particle



the PBD framework:

$$C(\mathbf{x}) = \frac{\rho_i}{\rho_0} - 1. \tag{2.8}$$

Note that ρ_i can be computed from the SPH density equation (Equation 2.10) and ρ_0 is the rest density of the fluid. This constraint could be projected in the same fashion as shown by the distance constraint. It is recommended to use constraint force mixing (RUSSELL, 2007) to regularize the constraint. This leads λ to be:

$$\lambda = -\frac{C_i\,(x)}{\sum_j |\nabla C_j\,(x)|^2 + \varepsilon}. \tag{2.9}$$

being ε a relaxation constant (MACKLIN; MÜLLER, 2013) specified by the user at the beginning of the simulation.

Liquids: As introduced by Macklin and Müller (2013) (MACKLIN; MÜLLER, 2013), a straightforward way to simulate fluids using a position based approach, is to introduce the calculation of the density, based on the Smoothed-particle hydrodynamics technique. Equation 2.10 defines how the neighboring particles j influences the interest particle i based on the kernel $W(p, r)$ where the radius r is a threshold, particles outside this threshold do not influence our interest particles. So, to introduce fluid particles, we need to apply the density constraint over all these particles.

$$\rho_i = m \sum_j W\left(\mathbf{p}_i - \mathbf{p}_j, r\right). \tag{2.10}$$

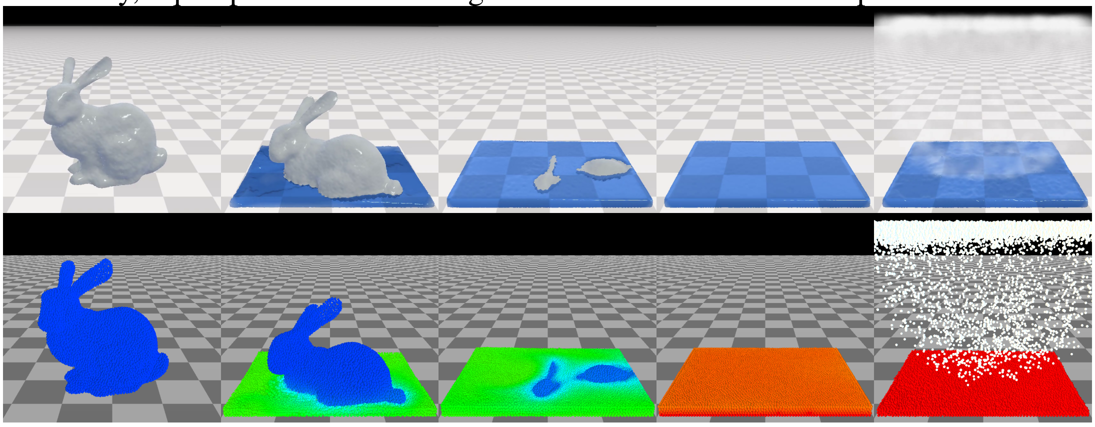
In this equation m represents the mass of the particle and $\mathbf{p}_i - \mathbf{p}_j$ the distance of a given neighbor to the interest particle. The impact of the neighbor on the calculated particle depends on this distance and the type of kernel applied.

2.2 Building a Virtual World: Simulating natural phenomena

This section is based on the following publication:
Steeven Villa Salazar, Jose Abel Ticona, Rafael Torchelsen, Luciana Nedel, Anderson Maciel. Heat-based bidirectional phase shifting simulation using position-based dynamics. in *Computers & Graphics Volume 76, November 2018, Pages 107-116*

Phase-change phenomena are present in our daily life. Examples are the evaporation of a fluid when it reaches its boiling temperature, the condensation of water vapor in air due to the pressure changes or due to the difference of temperature in boundaries, and the melting of snow when winter is ending. Current development in physics-based animation allows the simulation of these phenomena, but an integrated solution for modeling bidirectional phase-shifting objects is not available for games and other virtual environments. In this work we present a temperature-based method that drives phase transition phenomena based on latent heat of materials using position-based dynamics (PBD). Modifications to density, viscosity and distance PBD constraints are proposed to simulate the necessary thermal phenomena. Results show that melting, fusion, evaporation, condensation, dilation and even convection effects can be obtained by modifying the original PBD constraints in function of latent heat.

Figure 2.2: Melting and evaporation process with 27K particles involving three states and two phase changes: In the left image, a solid bunny is shown. When temperature increases at the bottom, the bunny starts to melt. After that, all particles are liquid and continue to heat. Finally, liquid particles switch to gas when the latent heat of evaporation is reached.



2.2.1 Related Works

Several models have been developed to better model the energy transport within this thermal problem (melting, fusion, evaporation, condensation) (GAO et al., 2017b; STOMAKHIN et al., 2014; HOCHSTETTER; KOLB, 2017; FARROKHPANAH; BUSSMANN; MOSTAGHIMI, 2017). Nonetheless, most of those works do not provide a general solution and focus on resolving specific parts instead of modeling the whole problem, usually covering one or two of the transitions. A common approach to achieve thermal and phase change simulations is to combine some of the existing techniques used to represent fluids (Smoothed Particles Hydrodynamics (SPH), Lattice Boltzmann Method (LBM), Fluid Implicit Particles (FLIP), Particle In Cell (PIC)) with techniques used to simulate solid bodies (Mass Springs systems (MSD), Finite Elements (FEM), Position-based Dynamics (PBD)). Those methods are not fully compatible and need special procedures to be coupled properly.

Melting and Fusion

In 2014, Alexey Stomakhin et al. (STOMAKHIN et al., 2014) published a work addressing melting, solidifying, and heat transfer using fluid implicit particles (FLIP) and Material Point Method (MPM) to simulate fluid particles. They obtain those effects varying material properties in function of temperature and phase, but their work does not address evaporation or condensation. Similarly, Gao et al. (GAO et al., 2017a) reported a work coupling fluid-solid phases using PBD and FLIP. However, they covered only melting simulation. In another work, Gao et al. (GAO et al., 2017b) extended their method to include gas, but they do not take into account condensation. Gao and colleagues were looking for a general solution to integrate the main phase-transition and states of the matter. We pursue very similar goals. However, we follow a different approach. The main difference is our fluid model. While they used a combination of Lagrangian and Eulerian approaches, we propose a unified fluid model to all states of matter. Very recently, Weiler et al. (WEILER et al., 2018) developed an implicit viscosity solver for Smoothed Particle Hydrodynamics (SPH) fluids capable to represent high viscous fluids consistently. Despite their work is mainly focused on viscosity handling, they extended the approach to simulate melting.

Evaporation and Condensation

Fluid-fluid interaction was modeled by Müller et al. (MÜLLER et al., 2005b). They reported a technique to simulate vaporization effects depending on the temperature of particles. Their work does not take into account other phase transitions and does not use latent heat as a criterion to induce the phase change. More recently, Hochstetter and Kolb (HOCHSTETTER; KOLB, 2017) published a paper simulating evaporation and condensation transitions. Their work is mostly focused on the visual effects. They used SPH as well as the Cleary and Monaghan Heat model (CLEARY; MONAGHAN, 1999) to manage particle temperatures. This is one of the models that we implemented in this work (see Sec.2.2.2). Ren et al. (REN et al., 2016) worked in a method to simulate gas using SPH and performing calculation just in the visible particles, unlike the traditional technique that consists on filling the domain with transparent air particles to generate convective forces. With the exception of the aforementioned papers, few works address both condensation and evaporation transitions using Lagrangian models.

Fluid-Solid Modeling

SPH is one of the most widespread frameworks to simulate fluids (IHMSEN et al., 2014b). To enforce incompressibility in SPH, Solentanthaler and Pajarola (SOLENTHALER; PAJAROLA, 2009) used a predictive-corrective approach, while Ihmsen et al. (IHMSEN et al., 2014a) presented an implicit incompressible SPH. Another relevant work addressing this topic was done by Becker and Teschner (BECKER; TESCHNER, 2007), where they introduced the widely used Weakly Compressible SPH. In 2013, Macklin and Müller (MACKLIN; MÜLLER, 2013) introduced the Position-based fluids, which is an implementation of SPH into the PBD Framework. Their approach helps to obtain an incompressible fluid at relatively low computational cost, consequently allowing large timesteps. Keiser et al. 2005 (KEISER et al., 2005) modeled solid and fluids merging Navier Stokes equations of movement into a Lagrangian framework. Despite SPH being mainly used to represent fluid models, the techniques have also been applied to model soft bodies (GRAY; MONAGHAN; SWIFT, 2001; BECKER; IHMSEN; TESCHNER, 2009).

In recent years, Position-based dynamics has become more and more popular. In 2014, Macklin et al. (MACKLIN et al., 2014) reported a “Summary” work demonstrating the versatility of PBD to simulate solids, fluids, plastic deformations, soft solids, and cloth. There is, nevertheless, a gap in the literature that our methods here presented aim to fill: providing the means to simulate plausible phase-shifting materials into a full Lagrangian framework.

2.2.2 Methods

In this section, we introduce a novel combination of methods and technologies into an integrated solution to cover all state changes in the same framework. In this sense, we chose and take advantage of the Position-based dynamics framework and its inherent benefits to model solid bodies together with the Position-based fluid approach to model liquid and gas. We then combine PBD and PBF particles with an SPH-based temperature model to track heat flow among them. For the sake of clarity, we marked our equations with a star ($\star$) beside the equation number to differentiate them from the other equations previously available in the literature.

Solid body dilation

Young's modulus ($E = \sigma/\epsilon$) defines the linear relationship between applied stress (σ) and associated deformation (ϵ) (MCNAUGHT; MCNAUGHT, 1997).

Elastic properties of solids can be introduced easily by extending the standard distance constraint of PBD (BENDER; MÜLLER; MACKLIN, 2015). We adapt this parameter as follows.

The more the temperature increases, the more the solid body dilates. Consequently, E must decrease as it represents the solid's rigidity:

$$E = \frac{2E_0}{1 + e^{\Delta T}}. \tag{2.11}$$

where E_0 is the Young's modulus of the material, a given property mostly obtained empirically. In addition, we establish the initial distance between particles to depend on the linear coefficient of expansion (e) and ΔT. This set up admits both dilation and contraction, depending on the direction of the heat change:

$$\Delta d_0 = e \Delta T d_0. \tag{2.12}$$

Hence, we obtain the final position correction for each couple of particles by adding (2.11) and (2.12) to the original corrections $C(x_i, x_j) = \mid x_i - x_j \mid - d_0$:

$$\left.\begin{aligned} \Delta x_1 &= -E\frac{w_1}{w_1 + w_2 + \tilde{\alpha}}\left[\mid x_{1,2} \mid - d_0\,(1 + e\Delta T) - \lambda\tilde{\alpha}\right] \\ \Delta x_2 &= +E\frac{w_2}{w_1 + w_2 + \tilde{\alpha}}\left[\mid x_{1,2} \mid - d_0\,(1 + e\Delta T) - \lambda\tilde{\alpha}\right] \end{aligned}\right\} \star\,. \tag{2.13}$$

Where d_0 is the initial distance.

Solid-Fluid-Gas Coupling

To handle boundary interactions, we calculate the density number $\delta_i = \sum_k W_{ik}$ according to the approach of Akinci et al. (AKINCI et al., 2012). Solid influence is taken into account while calculating the fluid densities. It is important to highlight that we correct the solid particle position by applying the density constraint to them. In such way, all phases interact in the same framework. Moreover, gas is treated as liquid-fluid particles but changing its internal properties. This will be discussed further in the section about transition coupling. The equation below shows how density calculation is performed:

$$\rho_i = \sum_j m_j W_{ij} + \sum_k m_k W_{ik} + \sum_b \frac{\rho_0}{\delta_b} W_{ib}. \tag{2.14}$$

where j particles are fluid (liquid and gas phase), k represents solid particles and b are boundary particles. So, i could be either a solid or a fluid particle. Using this approach, we can obtain solid-fluid-gas interaction in a straightforward fashion.

Heat Transfer

Temperature is the core variable in this work since phase change phenomena occur when a specific material reaches its temperature threshold (we will extend this assertion later in this section). In a homogeneous medium, heat transfer is given by (HOCHSTETTER; KOLB, 2017):

$$\frac{dT}{dt} = \frac{\nabla(k\nabla T)}{\rho c_p}. \tag{2.15}$$

The literature offers several SPH-oriented models using **heat** transfer. One of them is an explicit scheme to represent the time discretization of the heat equation (Equation 2.15). Besides, a 1999 work by Cleary and Monaghan (CLEARY; MONAGHAN, 1999) is still today one of the most popular models used to simulate **temperature**-based phenomena in SPH. So, we implemented both models to provide and track temperature distributions among our position-based particles:

Explicit Model: This is, perhaps, the simplest way to discretize Equation 2.15 to SPH:

$$\left(\frac{dT}{dt}\right)_{i,j} = 2\phi \sum_j \frac{m_j}{\rho_j} (T_i - T_j) \nabla W_{ij}. \tag{2.16}$$

Note that, in this equation, j is a neighbor particle independently of its current phase (excluding boundaries that have no phase information). The parameter ϕ is a scale factor to handle the amount of temperature transported among particles.

As boundary particles have no temperature properties, we handle this region in the following way:

$$\left(\frac{dT}{dt}\right)_{i,b} = 2\phi \sum_b \frac{\rho_0}{\delta_b} (T_j - T_b) \nabla W_{ib}. \tag{2.17}$$

This expression controls the influence of all boundary b particles around i. The total temperature in a single particle is given by the sum of the influences of the neighboring particles and the boundary heat, induced using Equation 2.17. This can be expressed as follows:

$$\left(\frac{dT}{dt}\right)_{i} = \left(\frac{dT}{dt}\right)_{i,j} + \left(\frac{dT}{dt}\right)_{i,b}. \tag{2.18}$$

Cleary and Monagham Model: C&M approach is an important reference on heat transfer related to SPH. They modeled heat transfer taking into account the specific heat capacity (c) and thermal conductivity (k), allowing for different conduction behaviors (CLEARY; MONAGHAN, 1999). They arrived at the following equation:

$$\left(\frac{dT}{dt}\right)_{i,j} = \frac{V_i}{m_i c_i} \sum_j \frac{4 k_i k_j}{k_i + k_j} V_j (T_i - T_j) \nabla W_{ij}. \tag{2.19}$$

In our modified model, we handle the boundaries by introducing fixed temperatures when evaluating boundary particles:

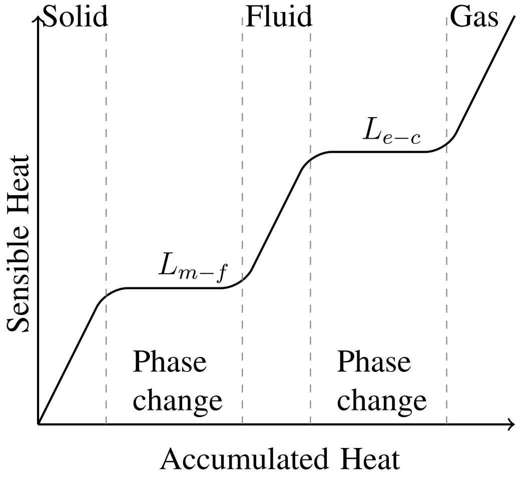
$$\left(\frac{dT}{dt}\right)_{i,b} = \frac{V_i}{m_i c_i} 2 k_i \sum_b \frac{\rho_0}{\delta_b} (T_i - T_b) \nabla W_{ib} \quad \star\,. \tag{2.20}$$

This allows us to induce temperatures coming from the boundaries.

Latent Heat

At the beginning of the previous section, we stated that phase change occurs when a specific material reaches its critical temperature. It is natural to think that a phase change depends on the temperature value and this is not completely wrong. The point is that, when the phase change is happening, the temperature remains constant. So, it is not the variation in temperature but the variation in energy that makes phase-change to occur.

Figure 2.3: Relation between heat and phase change



Let us define temperature as sensible heat, the heat we can measure with a thermometer. Latent heat, in turn, is the energy used to perform the phase change. We illustrate this in Fig. 2.3. In the initial (low temperature) state, a small amount of energy is stored. As the temperature increases (sensible heat), accumulated heat also increases. When the body meets a critical temperature (melting and boiling points), accumulated heat (latent) still increases but the temperature remains constant.

This means that the sensible heat stopped increasing. After gaining enough energy, i.e. after accumulating enough latent heat, the material reaches its latent heat of fusion/vaporization threshold and transitions to the next state of matter. Then, the sensible heat starts increasing again. This cycle is repeated in any of the phase changes and works in both ways. Notice that the latent heat quantity required to melt some material is rarely the same that is required to boil.

Computational implementation: To handle states and transitions, we store current state and last state depending on the particle temperature and the critical temperature values:

$$State = \begin{cases} Solid & T_i < T_m, \\ Phase\ change_{m-f} & T_i = T_m \\ Fluid & T_m < T_i < T_e\ . \\ Phase\ change_{e-c} & T_i = T_e \\ Gas & T_i > T_e \end{cases} \tag{2.21}$$

Sub-index $m{-}f$ and $e{-}c$ means $melting{-}fusion$ and $evaporation{-}condensation$ respectively. Here we cover all the possible states of our particles and, if a given particle is shifting its state, the temperature must be constant. The model saves the latent heat of each phase change independently: L_{m-f} for latent heat of melting and L_{e-c} for latent heat of evaporation. The rate of change in latent heat is calculated from the change in temperature and specific heat as:

$$\frac{dL}{dt} = c_i \Delta T. \tag{2.22}$$

with ΔT obtained from the heat model. Next, we increment latent heat applying Algorithm 1.

Our heat scheme works in the following way: when the phase change is occurring, which is to say $State = Phase\ change_{m-f}$ or $State = Phase\ change_{e-c}$, latent heat starts increasing or decreasing based on Equation 2.22, and then we enforce latent heat values to be within boundaries $[0, L_{threshold}]$. Other cases check if future temperatures are out of the current state and assign a phase change identifier to the particle.

2.2.3 Transition Coupling

Latent heat is the driver of all phase changes. Transition coupling is performed as shown in Algorithm 2. Further details will be given afterwards. Let us now overview the main loop. We start initializing variables in lines 2-4 for each particle. The values of these variables depend on material properties, and interactions between different materials are allowed. In line 6, positions are calculated according to the position based dynamics loop. Next (line 8), gravity values are assigned depending on the latent heat of vaporization.

Algorithm 1 Latent Heat Update

```
 1: switch $State\ Particle\ i$ do
 2:     case $Phase\ change_{m-f}$
 3:         $L_{m-f} \leftarrow L_{m-f} + \Delta L$ (Using Equation 2.22)
 4:         if $L_{m-f} < 0$ then
 5:             $L_{m-f} \leftarrow 0$
 6:         else if $L_{m-f} >= L_{m-f,threshold}$ then
 7:             $L_{m-f} \leftarrow L_{m-f,threshold}$
 8:     case $Phase\ change_{e-c}$
 9:         $L_{e-c} \leftarrow L_{m-f} + \Delta L$ (Using Equation 2.22)
10:         if $L_{e-c} < 0$ then
11:             $L_{e-c} \leftarrow 0$
12:         else if $L_{e-c} >= L_{e-c,threshold}$ then
13:             $L_{e-c} \leftarrow L_{e-c,threshold}$
14:     case $Solid$
15:         if $T_i^{n+1} \geq T_{m-f}$ then
16:             $State_i \leftarrow Phase\ change_{m-f}$
17:     case $Fluid$
18:         if $T_i^{n+1} \leq T_{m-f}$ then
19:             $State_i \leftarrow Phase\ change_{m-f}$
20:         else if $T_i^{n+1} \geq T_{e-c}$ then
21:             $State_i \leftarrow Phase\ change_{e-c}$
22:     case $Gas$
23:         if $T_i^{n+1} \leq T_{e-c}$ then
24:             $State_i \leftarrow Phase\ change_{e-c}$
```

From lines 11 to 15 all constraints are solved, then in line 17, we apply an algorithm to manage distance constraints on solid particles. Finally, we perform the heat transfer among particles. The last three lines are the updates on changed values.

Solid-Liquid Phase

When in the solid state, material properties are governed by the temperature value. Young's modulus E and α values are modified according to the current temperature (see Equations 2.12 and 2.11) until it meets the melting point. There, while the latent heat increases, both values remain constant. After that, the material attains the latent heat of fusion and its maximal thermal expansion, and consequently, the phase change occurs:

Solid-liquid (melting) Solid particles have distance constraints to conserve the solid's shape. Each particle represents a node in the solid and each constraint involves two particles. As a particle can admit more than one constraint, if its latent heat reaches the critic

Algorithm 2 Main Loop

```
for all Particles do
    $x_i \leftarrow 0, v_i \leftarrow 0, w_i \leftarrow \frac{1}{m_i}$
    $T_i \leftarrow T_0, L_m \leftarrow 0, L_e \leftarrow 0$
    Set $T_m, T_e, L_{m,threshold}, L_{e,threshold}, c_i, k_i, \phi_i$
loop
    for all Particles do Calculate Next Positions
    for all Fluid Particles do
        Set Gravity Acceleration($L_e$)
    (Using Equation 2.23)
    for all Particles do Neighborhood Search
    loopIterations:
        for all Particles do
            Density Constraint($L_e$)
        (Using Equation 2.8 and 2.24)
        for all Fluid Particles do Viscosity Constraint
        (Using Equation 2.7)
        for all Solid Particles do
            Distance Constraint($T_i$)
        (Using Equation 2.13)
    for all Particles do
        Manage Constraints($L_m$)
    (Using Algorithm 3)
    for all Particles do Heat Transfer
    (Using Equation 2.18 and algorithm 1)
    for all Particles do Update Temperatures
    for all Particles do Update Velocities
    for all Particles do Update Neighborhood
```

heat, all constraints containing this particle are deleted.

Liquid-solid (freezing) In contrast with melting, freezing (or solidification) does the opposite process. Here, we create new constraints based on a support radius rs_{max} (not necessarily the same used with the SPH kernels). When two neighboring fluid particles are decreasing their latent heat, they are candidates for solidification. If both meet the solidification critical value, a new distance constraint is generated. This process stops as soon as the particle is out of the $Phase\ change_{m-f}$ status. Notice that, for a particle to solidify, its last state must be *fluid*. Additionally, if a particle reaches its maximum number of constraints nC_{max}, no new constraint are created for this particle. Algorithm 3 is introduced to manage the life cycle of distance constraints.

Algorithm 3 Manage Constraints

```
for all Distance Constraint Particles i,j do
    if $State_i = Phase\ change_{m-f}$ & $State_j = Phase\ change_{m-f}$ then
        Delete Distance Constraint i,j
for all Particles i do
    if $State_i = Phase\ change_{m-f}$ then
        for all Neighboring Particle j do
            $Cond\ 1 \leftarrow State_j < Phase\ change_{m-f}$
            $Cond\ 2 \leftarrow \mid x_i - x_j \mid < r_{max}$
            $Cond\ 3 \leftarrow nC_i < nC_{max}$
            if $Cond\ 1$ & $Cond\ 2$ & $Cond\ 3$ then
                new Distance Constraint i,j
```

Liquid-Gas Phase

To obtain vaporization, we modify the initial rest density ρ_0 in the density constraint (Equation 2.8) multiplying by a scaling function β. Doing this, convection behavior similar to water boiling arises when latent heat is increasing. Equation 2.23 represents an inverse sigmoid function:

$$\beta\left(\gamma\right) = 1 - \frac{1}{1 + \left(1 + \frac{12}{L_{e,threshold}}\right)^{\gamma}} \quad \star\,. \tag{2.23}$$

with $\gamma = \frac{L_{e,threshold}}{2} - L_e$. The behavior of Equation 2.23 is shown in Fig. 2.4. Moreover, vaporization simulation is performed setting the particle gravity as a function of the latent heat. The function g (shown in Equation 2.24) keeps the particle gravity unchanged in smaller values of latent heat and, as it increases, gravity turns to positive, producing the vaporization effect (see Fig. 2.4):

$$g\left(\omega\right) = \frac{10.8}{\ln\left(\frac{L_{e,threshold}+0.02}{0.02}\right)} \ln\left(\omega\right) - 9.8 \quad \star\,. \tag{2.24}$$

with $\omega = \frac{L_{e,threshold}+0.02}{L_{e,threshold}+0.02-L_e}$.

2.2.4 Results

We implemented our model in C++ with the Eigen library. We used a 3.40 GHz i5-4670 CPU with 16GB RAM to generate all results here presented. The whole implementation is sequential on the CPU for the sake of simplicity and reproducibility. Besides,

Figure 2.4: Phase Functions: Buoyancy of a particle is given by this function. This allows us to simulate ebullition and evaporation effects. Blue: (Right scale) The density factor decreases as latent heat increases, providing a natural convection alike effect. Red: (Left scale) Gravity remains low in almost the 90 percent of the phase change, letting the convection effect occur. Gravity arises when the density is low generating the evaporation effect.

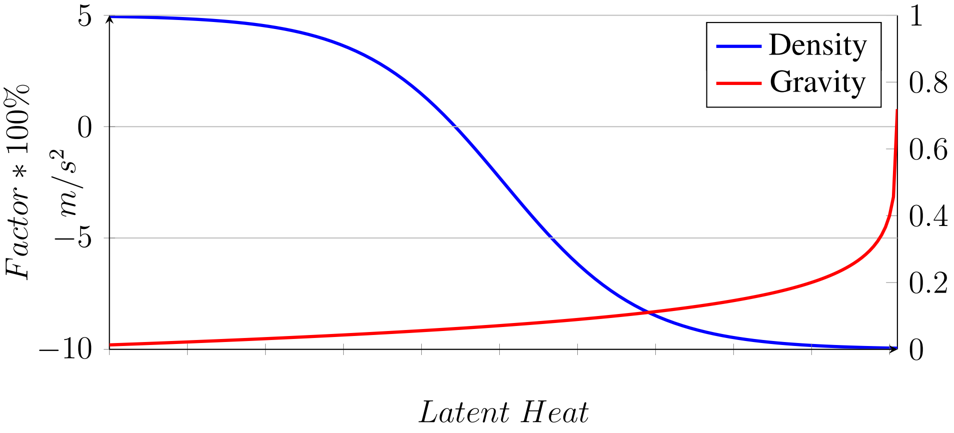
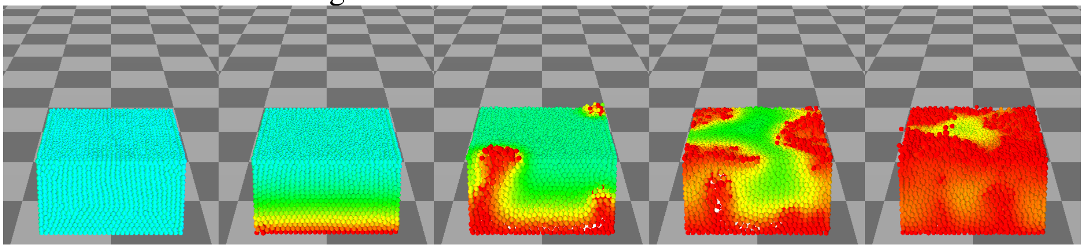


Figure 2.5: Natural Convection effect



we used post-processed rendering to better isolate the software elements for analysis. The Nvidia Flex environment was used for graphics rendering.

To assess the capabilities of our method, we developed scenarios that cover the effects of all phase transitions.

Scenarios

Melting and Vaporization: Fig. 2.2 illustrates the melting and vaporization processes with 27k particles. We set a fixed high temperature on the floor where the solid-cold bunny is placed sitting. When a particle meets the latent heat threshold, our constraint manager breaks all the connections among this particle and its neighbors. When the particles change their states to liquid, the viscosity constraint is activated. Later, the transition to gas completes the process. This scenario shows one way of the whole process, involving three states: solid, liquid, and gas, as well as the two phase changes.

Figure 2.6: Melting simulation: Hot water drops over an ice-bunny that melts when hot particles contact with cold particles. Interestingly, some liquid particles cool enough to freeze when in contact with the very cold ice particles.

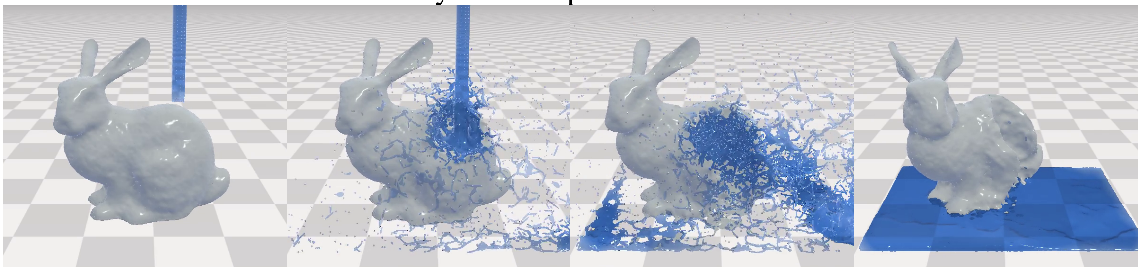


Figure 2.7: Ebullition (left) and Condensation (right) effects. Water in ebullition, with 50k particles, demonstrates the effect of vaporization combined with convection that is typical of boiling water. With 26k particles, condensation illustrates the cloud formation and precipitation (rain).

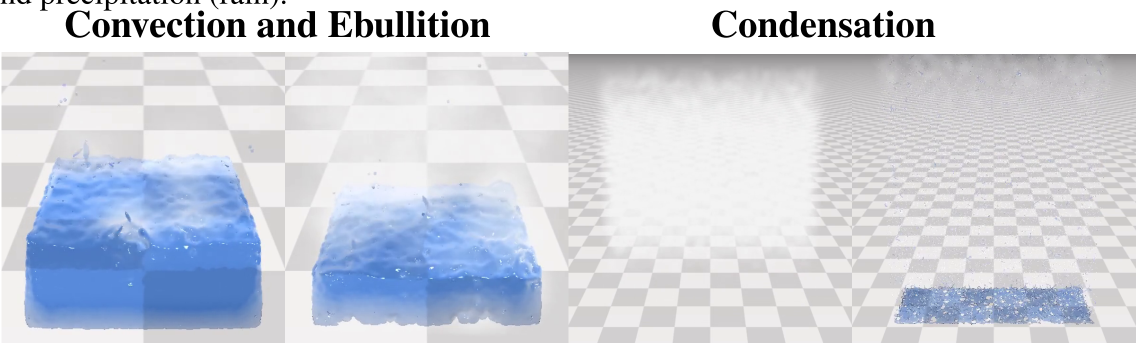
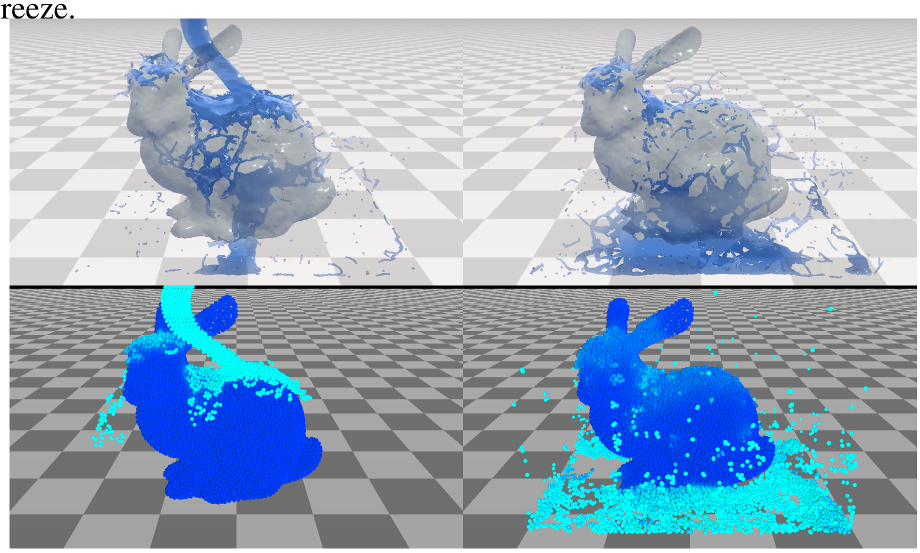


Figure 2.8: Solidification simulation: Warm water falls over the icy bunny but the fluid particles have not enough heat to melt it. Then some fluid particles lose enough of their heat to freeze.



Solid and Fluid Interactions: When a cold solid is thrown into a warm liquid, or vice-versa, the heat-transfer phenomena proceeds to eventually modify the phase of some of the particles in both the liquid and the solid, e.g. the interaction between ice and warm water. A higher amount of energy is present in the hot liquid particles. Then, energy is transported from the liquid, let us say, water, to the cold solid, melting it and decreasing

Table 2.1: Average times per frame in milliseconds for each scenario. NS: Neighborhood search, CP: Constraint projection, VS: Viscosity, HT: Heat transfer, CM: Constraint manager

Scenario	Particles	Total	NS	CP	VS	**HT**	**CM**
Heat transfer	18k	59.43	6.38	45.047	3.82	3.68	3.66×10^{-2}
Vaporization	18k	60.65	8.64	44.50	3.31	3.64	3.79×10^{-2}
Condensation	20.4k	64.35	15.46	41.33	3.51	3.46	3.96×10^{-2}
Melting	34.6k	409.66	20.37	353.01	6.10	6.32	22.89
Solidification	33.5k	218.46	16.66	179.32	4.67	5.23	11.66
Full Simulation	27.1k	97.39	14.35	71.71	4.35	5.13	1.06

the sensible heat present in the remaining water. Fig. 2.6 depicts a scenario where some amount of hot water falls over an icy bunny, melting the ice partially in the regions affected with enough energy to break the local solid constraints. When the falling particles do not have enough energy to break the constraints, however, a different phenomenon takes place. In the scenario of Fig. 2.8, instead of melting the bunny, some liquid particles eventually solidify when they lose enough heat.

Natural Convection: The use of sigma and logarithmic functions allows us to simulate convective effects on fluids. In a boiling water scenario, the change in density due to the increase of latent heat causes particles with higher energy to move up and particles with less energy to move down. In Fig. 2.7 a boiling effect is shown in the top frame. When

Figure 2.9: Computational cost of each functionality according to the number of particles in the model. Notice that the cost for the Heat Transfer and the Constraint Manager are fairly lower than the Constraint Projection (PBD solver) and also grow much slower when the number of particles increase.

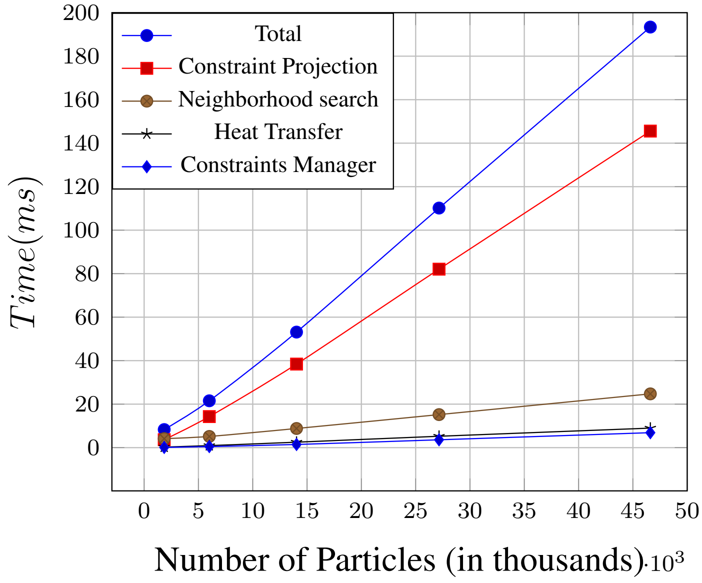
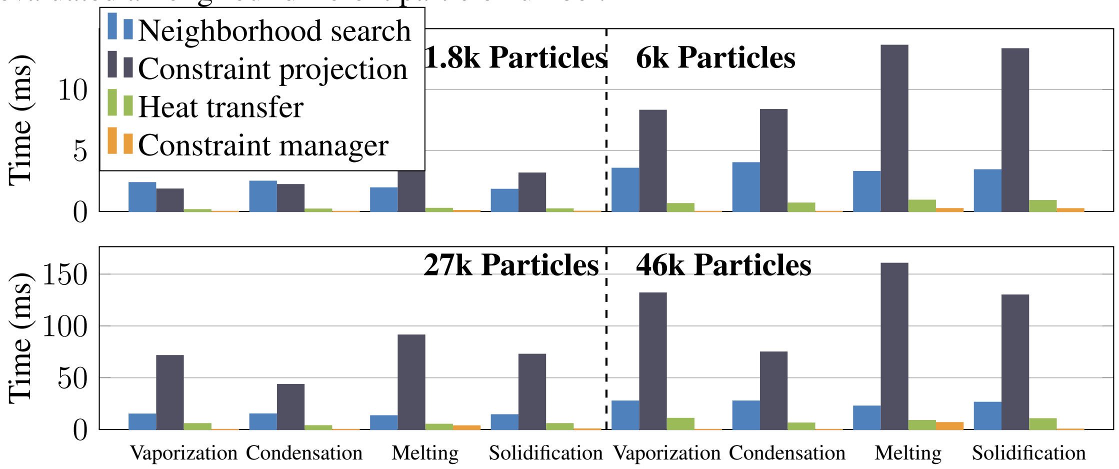


Figure 2.10: Performance averages at constant number of particles. Four scenarios are evaluated among four different particle number.



lower particles absorb enough energy, they rise to the surface and eventually vaporize. Figure 2.5 shows explicitly how particles with lower density and higher temperature arise from the bottom to the surface of the container.

Vaporization & Condensation: In a similar way to the convection scenario, we use the σ function to simulate vaporization. The buoyancy of gas particles depends on their latent heat of evaporation. The more latent heat of evaporation a particle has, the more buoyancy it experiences. Other works such as the ones proposed by Muller et al. (2005) and Macklin and Muller (2014) (MÜLLER et al., 2005b; MACKLIN et al., 2014) suggest the introduction of air particles in the free domain to generate buoyancy and let the gas particles raise. Such approach could help to better model the fluid-air interaction in terms of heat transfer. However, its computational efficiency is questionable under some circumstances. Fig. 2.2, in its last two frames, shows our evaporation process in action: fluid particles start a convective-like effect and when enough energy is available, buoyancy effects begin raising these particles based on Equation 2.24, as Fig. 2.4 illustrates. Then, in Fig. 2.7, both effects are shown. The bottom row frames show the condensation effect. The simulation starts with vapor particles in the upper side of the container. As they meet a very cold surface at the top of the box, vapor particles gradually transition to liquid and fall back.

Performance analysis

Table 2.1 show the performance of our approach in seven different scenarios, summarizing the computational impact for each functionality. *Total* shows the total average time per simulation step. For this performance test, we computed 8 PBD internal iterations per step. The table shows that constraint projection takes longer than all other functions. As a reminder, constraint projection is a fundamental step in PBD. An interesting point here is the relatively low impact of our methods, the heat-transfer model and the constraint manager, on the overall simulation. Heat transfer computation takes less than 8.8% of the total time in the worst-case scenario (**Heat transfer** row). Excluding that scenario, the mean value is even lower: 5.64%. Besides, our *constraint manager* takes even less time than the heat transfer. In most cases, the execution time was under 1% of the total cost of a timestep. These results indicate that the additional computational cost imposed by our methods is barely noticeable in relation to the original framework without phase changes at all. It is important to highlight that our method was implemented on the basis of Jan Bender's sequential code (BENDER; MÜLLER; MACKLIN, 2015), available on his official website[1].

Particle number analysis

We analyzed how the computation time grows with the number of particles. Fig. 2.9 compares the average computation time per step of simulation for each functionality. For this test, we took the melting demo and varied the number of particles in the bunny model. The chart displays a linear behavior for times in relation to particles number. Visibly, constraint projection takes most of the total time of each timestep. Heat transfer, constraint manager and neighborhood times search times also grow linear with the number of particles, but their growth factor is much lower. This indicates that the percentage of phase-shifting cost in relation to the total cost tends to decrease for larger numbers of particles. In Figure 2.10 we compare the computation time among simulations when they have the same number of particles. At a very low number of particles, the neighborhood search is more expensive than the constraint projection (for Vaporization and Condensation). On the other hand, among the other scenarios and particle configurations, the constraint projection takes more time of the total timestep. Despite that, even with larger number of particles the neighborhood search time remains stable and the constraint manager has no noticeable influence in vaporization and condensation effects. The hardest simulation was melting, which took more time independently of the particle number. This was the scenario where our constraint manager had the worst behavior. Nevertheless, the constraint manager has lower impact on the whole simulation, even at a large number of particles.

[1] http://www.interactive-graphics.de/

Table 2.2: Parameters used along the simulation

Notation	Description	Range	Value
E_0	Initial Young's modulus	$(0-1]$	1
e	Linear coefficient of expansion	$[0.0-0.01]$	0
nC_{max}	Maximum constraint number	$4-10$	4
rs_{max}	Maximum radius for solidification	$[2r-4r]$	0.055
h	SPH Support radius	-	4r
c	heat capacity	-	1000
$T_{melting}$	Melting Temperature	-	0
$T_{evaporation}$	Evaporation Temperature	-	100
L_{m-f}	Latent Heat melting - fusion	-	333
L_{e-c}	Latent Heat evaporation - condensation	-	2257
k	Thermal conductivity	-	Depends on the demo
ϕ	diffusion of particle	$[10-270]$	Depends on the demo
μ	viscosity	$0.025-1$	0.025
r	Particle radius	$0.01-0.025$	0.025
ρ_{solid}	density of solid particle	$[700-1000]$	900
ρ_{fluid}	density of fluid particle	$[700-1000]$	1000
ρ_{gas}	density of gas particle	$[700-1000]$	700
m	mass of particle	$[0.0064-0.07]$	$0.8*8r^3*\rho_{solid}$
α	Compliance	$[0-1]$	$0.16e-9$

Parameters and Method Stability

Table 2.2 shows the principal variables used along the simulations. As most of them were mentioned on the body of this document, in this section we will just address these parameters regarding the methods stability and values we selected for the simulations.

First, the final value of the distance constraint relies on Young's Modulus E_0 and Linear coefficient of expansion e (listed on section 3.1). The value of E_0 must be within the 0 to 1 range because the temperature based E uses this value and an inverse sigma function so that it acts like a damper. Larger values of E_0 lead to instabilities in the solid phase from the first timestep. The main difference between E_0 and α from XPBD is the variation regarding the current temperature of the particles. Likewise, the parameter e is constrained to the range from 0.0 to 0.01. When set to 0, there is no dilation generated from temperature, and the bigger this value the faster a body dilates and contracts. Values above 0.01 cause instabilities because of the velocity of contraction and dilation.

The next set of parameters affects the freezing behavior. nC_{max} controls the maximum number of constraints that one can attach. Despite any value above 0 could be set, the number of constraints highly affects the consistency of solids. Particles with less than 4 distance constraints are quite deformable. Particles with more than 10 constraints turn the simulation unstable because our constraint manager does not create distance constraint with a predefined structure, so particles within the maximum radius for solidification rs_{max} and meeting the solidification temperature will be connected instantaneously with distance constraints. Moreover, rs_{max} values under $2r$ will not create any new constraint due to the minimum distance between particles. Values above $4r$ are suitable to simulate but the behavior of the new solid is somewhat unpredictable: distance constraints very different within the same body cause inconsistency when the body is contracting or dilating.

The third set of parameters in Table 2.2 are the thermal properties of the material. Those values mainly affect the time for the heat to move from a particle to another. Heat capacity (c) directly influences the amount of latent heat in a given particle (Equation 2.22). So, this value changes the velocity of phase change. We used a constant value of 1000 in our simulations but changing this value will not affect the simulation stability. Values for $T_{melting}$, $T_{evaporation}$, L_{m-f} and L_{e-c} were constant as well, and we used values corresponding to water. Since in real world $T_{melting} < T_{evaporation}$, to meet this inequality is the only restriction to set those values. An incorrect selection of threshold temperatures will lead to artifacts because particles with distance constraints will arise, compromising the stability of the model. Moreover, L_{m-f} and L_{e-c} together with c define the time that a phase change takes in the simulation. Again, an incorrect value selection will not affect the stability but it is recommended to use real world values to obtain a simulation within a reasonable phase change time. Thermal conductivity (k) and ϕ are closely related in the sense that those variables define the heat transfer velocity on each of the schemes: Explicit or Cleary and Monaghan. Selecting the correct values for either of the models should cause the same behavior, but the Cleary model advantage is the capability to work with different materials interactions. The final value depends on the k value of each neighbor particle, while in the explicit model, ϕ is the same for each neighbor particle. Thus, it is not the selection of heat model but the parameter of each model that affects the behavior of the simulation.

Discussion and Limitations

In this sequential implementation of PBD, simulations up to 8k particles are interactive. Above this number, the framerate drops to non-interactive frequencies. In comparison with the basic PBD code, our approach increases the computational cost by only 8% in the worst case. In current optimized parallel implementations of PBD, such as that of the Flex library, models containing near 100k particles can reach interactive framerates. This indicates that an eventual integration of our algorithms with a GPU implementation of PBD, e.g. using CUDA, may increase performance by at least one order of magnitude.

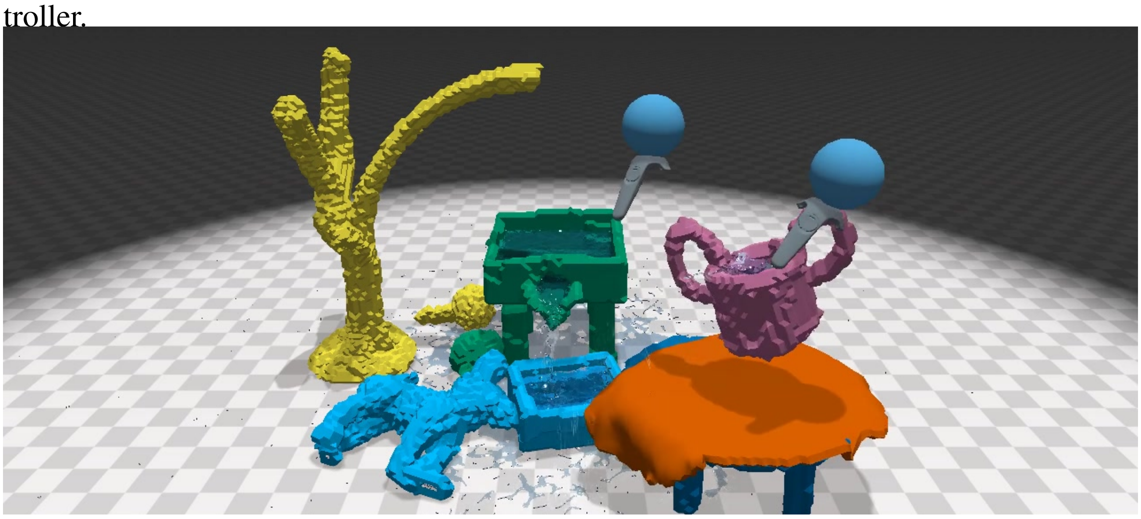
In all our tests we ignored heat loss due to particles interaction with the atmosphere. This leads to inaccurate behavior in some situations. For instance, when gas particles are raising from a boiling fluid, they should lose their energy when in contact with air (or another gas) particles in the atmosphere. Moreover, as we simply use distance constraints to simulate solids, we do not achieve rigid body behaviors. Instead, our solid bodies allow deformation. This could be improved by introducing more consistent constraints to the constraint manager. A cluster approach to model rigid elements, as available in the Flex library, should also be investigated further. As our constraint manager creates distance constraints when particles meet the two requirements (Temperature of both particles and distance between particles), the newly formed solids are not totally consistent. It is necessary to improve the constraint manager to create structured constraints depending on the information of the current constraints in the particle.

2.3 Immersing into the virtual world: Interacting with physics-based objects

This section is based on the following publication:
Jose Abel Ticona, Steeven Villa Salazar, Rafael Torchelsen, Anderson Maciel, Luciana Nedel. Phys-Sketch: Sketching 3D Dynamic Objects in Immersive Virtual Reality. in ***CGI 2019: Advances in Computer Graphics pp 119-130***

Sketching was traditionally a 2D task. It was later brought to the 3D paradigm when the new generation of VR devices allowed the end users to get in touch with this application. However, the dynamism of the drawn models has not changed when they moved to 3D. The models are static in the scene and do not have physical behaviors. Some previous works add kinematic animations or dynamic rendering as movement effects,

Figure 2.11: Different bodies interacting in the same VR scene and with the user controller.



but the core of the sketch remains static. In this section, we introduce a new physics-inspired sketching technique built on the top of Position-based Dynamics to enrich the 3D drawings with real-world behaviors. A particle-based method allows interacting in real time with a wide range of materials including fluids, rigid bodies, soft bodies, or clothes. Users can interact with the sketches and dynamically model or sculpt it. We present an exploratory usability test with users that demonstrate the capabilities of the approach.

2.3.1 Related Works

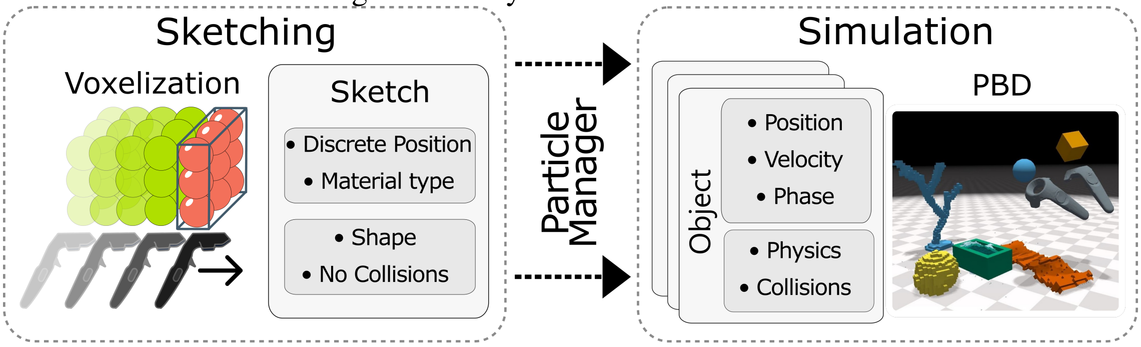
Lagrangian models are widely used to perform real-time simulations. Methodologies as Position-based dynamics (PBD) (MÜLLER et al., 2007) or Smoothed-particle hydrodynamics have become more and more popular lately. Some works were intended to model physical behaviors as solids, fluids (MACKLIN; MÜLLER, 2013), gases (REN et al., 2016) and even complex phenomena as phase transitions (As presented in section 2.2 (SALAZAR et al., 2018)) .The main applications of those methods are animation and films, but some works use them in interactive simulations: Pan et al. (PAN et al., 2015) simulated real-time soft tissue cutting with position based dynamics getting very plausible results, after this Berndt et al. (BERNDT; TORCHELSEN; MACIEL, 2017) introduced a faster simulation model. An interesting feature of both works is the use of force feedback, that requires even stringent computation times. Despite its capacity for real-time simulation, particle-based methods are mainly used to create animations or movies, and few works are reporting interactive applications. Lagrangian models in contrast with Eulerian approaches, treat the bodies as the origin of its calculations, giving the properties and states to each particle making unnecessary to calculate properties inside a grid, this is useful especially when the domain of the simulation is unknown. There are several meshless methods used to model continuum mechanics; we chose Position-Based Dynamics (MÜLLER et al., 2007) because of its unconditional stability and its low computational cost.

Some popular applications were recently proposed to allow 3D creations employing immersive interaction, such as the TiltBrush (TILT. . . ,) released in 2016 by Google. This application adequately addresses most of the immersive interaction issues when painting in 3D. In the same way, Quill (QUILL. . . , 2018), from Facebook, lets the user create and animate virtual models. These approaches were successful in allowing users on materializing ideas, since both introduced tools to sculpt, draw, and paint. In the case of Quill, it is also possible to add a timeline to animate frame per frame the scene. Within this paradigm, the user can choose presets to generate animations or create the scene and then animate it.

The more noticeable weakness on these tools is that it is impossible to choose a material for the objects. Some of the applications let the user associate materials to their models, but this is limited to rendering properties. Mechanical behavior remains the same.

Works as Canvox by Kim et al. (2017) (KIM et al., 2017) and Multiplanes by Barrera et al. (2018) (Barrera Machuca et al., 2017) focus on how the artist does the designs, the first one proposes the division of the whole canvas in smaller volumes of interest to give more details when drawing. In contrast, Multiplanes aid the user to sketch by automatically generating planes as the user draws a line. Seo et al. (2018) (SEO; BRUNER; AYRES, 2018) presented Aura Garden in 2018, a collaborative sculpting environment for light. This environment lets users draw and animate in mid-air with different materials, but all those materials are non-physical (excluding wood), so it is not possible to simulate interactions among them and with the user. Another Collaborative application in VR is Ontlus by Chen et al. (2018) (CHEN et al., 2018) where the users can paint and sculpt in the virtual environment. A common failure of those applications is the few or even inexistent physical-based animation on the process, limiting the dynamism of the creations. Thus, the interactivity of the objects decreases. Eroglu et al. (2018) (EROGLU et al., 2018) recently published a successful physic-based model to sketch in VR, they focused on fluid modeling, letting the user change the fluid properties and freely draw in space, although they did an outstanding work, their method is limited to fluids, and do not take into account solids or soft bodies.

Figure 2.12: Our approach constantly switches between two main states for each new body in the scene; Sketching and simulation. The first state is used to create new elements into the scene by placing new particles as long as the user moves the controller through space. Dynamic behavior is given to the bodies on the second stage depending on the properties of each particle. The dynamic of the previously created bodies is not stopped when the user starts drawing a new body.



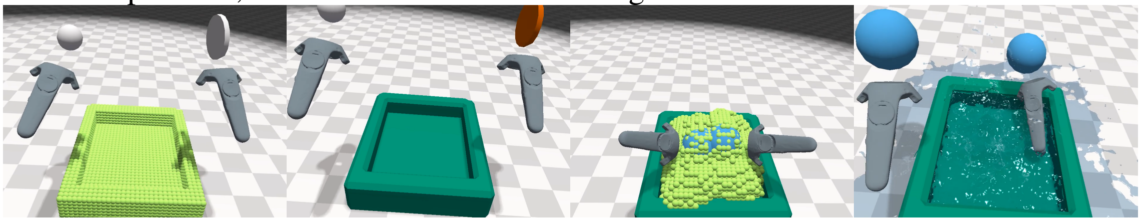
2.3.2 System Overview

Our approach is divided into two main modules. The first one is the *Sketching* module and is the creation stage, where the user draws objects. Objects are sketched taking into account its initial position and material type. Users are free to select a material and sketch objects in any achievable position and with the shape they want to have. During the sketching, however, for the sake of controllability, objects are not behaving according to their physics properties. For performance reasons they are represented as a set of spheres (see Figure 2.12 left). This process is called *Voxelization*. Section 2.3.3 explain in detail the processes involved in the sketching stage.

The second module of our application is the *Simulation* and occurs when the user indicates the *Sketching* of a given object is concluded. In fact, the simulaton stage involves the particle management: physics is introduced, and the bodies get their physical behavior (see Figure 2.12 left). In this stage, the objects sketched interact with the environment, with other objects and with the user, that can touch, stretch or move any of them as desired. Figure 2.11 shows a dynamic scene composed of five solid objects(two tanks, a table, a tree and a pot), four soft objects (two fruits, a dog and a cloth on the table), and two fluid objects (water inside the tank and tea inside the pot).

During the experience, the user can switch between the stages every time he draws

Figure 2.13: Step by step of Sketching-simulation stages: left image shows the sketching stage where particles are in the absence of any physic behavior to make easier the composition of the body. Second image shows the same object after the sketching stage, here the new body is now a solid and has physic properties. Third, shows the creation of a new body, this time as the brush is blue, the new body will be fluid. Note that the already created objects stills in the simulation stage. In the last image, fluid properties were given to the new particles, and both bodies are interacting in the same scene.



a new body.

Our system is built over the standard position-based dynamics (PBD) framework. Algorithm 4 gives a general view of one timestep of our simulation. After the main PBD calculations, we update both velocities and positions of the virtual tool using the external tracking system (see Section 2.3.3). When the user creates a new object (Line 5, from Sketching to Simulation stage), we capture the path traveled by the control and the material which was chosen by the user.

Then, this data (position, velocity, and phase) is introduced in the function *CreateParticles* (Line 6, Available materials are shown in Section 2.3.3), and next, depending on the material, the corresponding calculations are performed. Details of each calculation made between lines 8 to 15 (mass center, relative positions, cluster mass center, cluster relative positions, and density) can be also found in Section 2.3.3.

In line 16, once we have the material properties, the particles are moved to the main vector, and then we sync (Line 17) the new vector with the running simulation to avoid memory conflicts.

If the user is not creating an object but modeling it (Line 18, Sketching stage), here we use position and velocity of the tool to create a soft path (Line 19, Section 2.3.3). Having this path we can proceed to create or remove particles using algorithms 5 and 6 (see Equation 2.31).

Next, in lines 20-23 we define the boolean option to perform. If the selected material is *Solid*, *Soft* or *Fluid*, we create the voxelized particles which are candidates to become an object (Line 21). However, if the material does not correspond to one of the mentioned before, we delete the particles in the path. The process ends in line 24 when vector P_{shape} is finally rendered.

Algorithm 4 Main Loop

```
for all Particles do (Timestep Starts)
    PBD Core calculations
    $P_{Tool} \leftarrow UpdatePositions$ (Section 2.3.3)
    $V_{Tool} \leftarrow UpdateVelocities$ (Section 2.3.3)
    if $NewObject$ then
        $createParticles(P_{shape}, MaterialType)$
        switch $MaterialType$ do
            case $Solid$ (Sec. 2.3.3 Solids)
                $calculateMassCenter$ (Fig. 2.18)
                $calculateRelativePositions$ (Fig. 2.18)
            case $Deformable$ (Sec. 2.3.3 Soft, Clothes)
                $calculateClusterMassCenter$ (Fig. 2.19)
                $calculateClusterRelativePositions$ (Fig. 2.19)
            case $Fluid$ (Sec. 2.3.3 Fluids)
                $CalculateDensity$ (Fig. 2.1)
        $addParticles$
        $sync$
    else
        $P_{path} \leftarrow InterpolationPosition(P_{tool}, V_{tool})$
(Sec. 2.3.3)
        if $Material$ then
            $P_{shape} \leftarrow Voxelization(P_{path}, typeShape)$ (Alg. 5)
        else
            $particleRemotion(P_{path})$ (Alg. 6)
    $draw(P_{shape})$
    PBD Variable updating
```

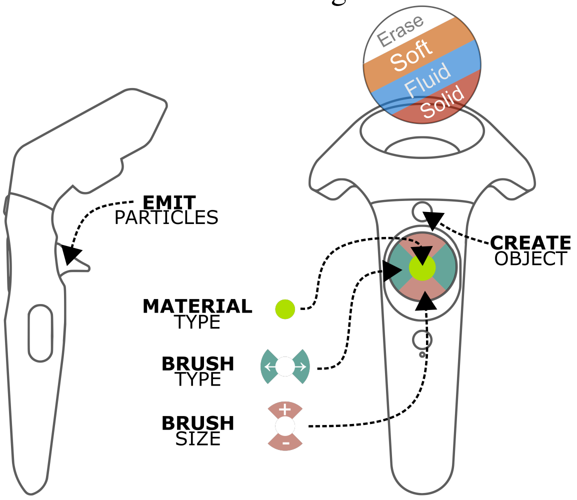
2.3.3 Methods

Sketching

To create an object within our framework, the user must follow two steps: First sketch the body and then giving it physical behavior. Although these two stages are well differentiated for the user when creating an object, the simulation keeps always running on background, and newly created objects are smoothly added to the simulation stack.

This separation of stages is necessary to make the sketching easier as long as the user can last as much as he wants. Without this differentiation the newly introduced particles would start falling since its creation, making it harder to create consistent sketches. In the first moment, the generated particles do not have any animation nor rendering (Except the primitive sphere). At this moment, the user needs to start generating particles by using the brush. The brush is the central user-environment interaction tool, which has three main functions: Create, delete and move objects. Though the brush is a platform-

Figure 2.14: Our method uses the HTC VIVE Controllers to track the user's hands. The image describes the possible interactions while sketching. The circle over the control represent the feedback received by the user while sketching, the colors of this sphere (or chosen shape) defines the behavior that will be given to the new body



independent generic tool, in this case, we have represented the virtual brush as an HTC VIVE controller wich holds the shape and color of the material on its top. Figure 2.15 shows both shapes and material type availability, the user can easily select the object properties through the controller buttons. Every time the user changes a property, visual feedback is provided on the top of the controller (As shown in Figure 2.14). The amount of particles introduced in the environment is given by the size and shape of the brush: New particles are evenly introduced (See Section 2.3.3) within the volume of the chosen form. The method to create objects in the scene is the movement of the brush through space, the travels of the tool create paths conformed of particles, and these paths are the future objects. In this way, to create a solid body, the user must manually fill each part of the object.

Particle creation can be done with both hands at the same time without overlap conflicts because we correct particle positions on the go. Two hands sketching is free, and any hand can create or delete particles.

Some issues could happen when the brush moves so fast, for instance some gaps in the path could appear and make it looks discontinuous, but the stability of the sistem is guaranted.

Figure 2.14 show the possible interaction through the controller: To emit particles the user must press the trigger and hold while the sketching, when the trigger is released no particles will be created but the object will remain static until the button *Create Material* is pressed. Between these two interactions, the user can modify the brush shape and size as well as the material type using the circular trackpad. Pressing the center of the trackpad

Algorithm 5 Voxelization

```
procedure VOXELIZATION($P$)
    $p = discretization(P)$ (Using Equation 2.28)
    if $p \in M$ & $M[p] < 0$ then
        $P' = interpolation(p)$ (Using Equation 2.29)
        $M[p] = index$
        $Particles[index] = P'$
        $index + +$
```

changes the material type, while the vertical axis the brush size and the horizontal the brush type.

Movement through the scene is allowed during all the simulation, enabling users to create entire compositions instead of only modeling individual objects. Even though the virtual space is not limited, there are some physical and computational constraints to the movement through the scene; User can move as long as the wires of the HMD let him move. Also, the size of the compositions is limited to the number of particles allowed in the scene, around 100k particles are easily managed in real-time simulations.

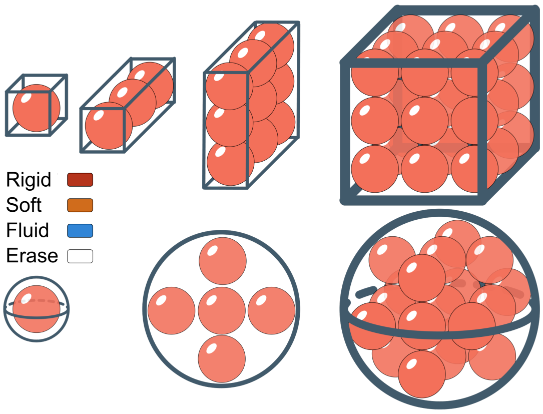
Voxelization

The way as the particles are introduced to the environment impacts the behavior of the simulation: The placement of more than one particle at the same position could make the particles move at high velocities, as we let the users freely introduce a high amount of particles each frame, an expansive wave effect could be expected. This issue does not imply an instability on the simulation because overshoots are avoided using PBD, but the first iterations right after the placement of the particles will have a non-physical behavior. To avoid this, we perform a voxelization in the sketching stage, where each voxel is a particle. The whole scene is divided in a binary 3D grid $M \in R3$ where the size of each cell is the size of a particle (Figure 2.16. M is initialized with -1 Values, when a body is created in a given space inside the scene, the values in the grid are marked as occupied (0 Values), this is especially helpful when deleting particles since we know the positions

Algorithm 6 Remove Particles

```
procedure PARTICLEREMOTION($P$)
    $p = discretization(P)$ (Using Equation 2.28)
    if $p \in M$ & $M[p] \geq 0$ then
        $indexRemove = M[p]$
        $M[p] = -1$
        $removeParticle(indexRemove)$
```

Figure 2.15: Voxelization: Basic particle creation shapes, On the top: The shapes created starting from a cube (Particle, line, plane, sphere). on the bottom: Shapes created starting from a sphere (Particle, circular plane, sphere)

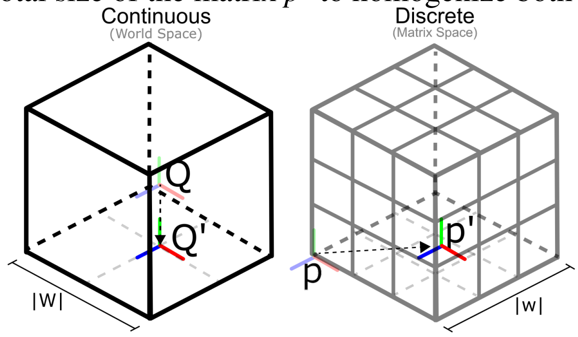


of the newly introduced particles inside the vector. If the user moves again the primitive to add more particles in a place already filled, no new particles will be introduced at that place. This proceeding is shown in algorithm 5 where the first step is the discretization of the positions. Then we check if the particle is actually within the range of M and whether the position is free or not, then we move the position again to the continuous space but this time the particles will be evenly arranged because of the rounding while the discretization, finally the matrix M is updated. To remove particles, the algorithm 6 is applied. We homogenize the coordinates of the space and the coordinates of our binary 3D grid using the following relations:

$$
\begin{align}
p' &= p - c \tag{2.25}\\
Q' &= Q - H \tag{2.26}\\
\frac{p'}{Q'} &= \frac{|w|}{|W|} \tag{2.27}\\
p' &= Q'\frac{|w|}{|W|} \notag\\
p - c &= (Q - H)\frac{|w|}{|W|} \notag\\
p &= (Q - H)\frac{|w|}{|W|} + c \tag{2.28}
\end{align}
$$

Where p is the original coordinate system (CS) of the 3D grid, p' is the new coordinate system modified by the center of the grid (c), and Q is the original coordinate system of the world, given by the configuration of the headset. This coordinate system is placed on the center of the workspace in a given distance to the floor; we move this coor-

Figure 2.16: Space Map: World space coordinates coming from the immersive environment starts at a given point in the middle of the workspace, we translate that point *q* to the floor *q'*, and coordinates from matrix starts at (0,0,0) point *p*, we translate that point to the middle of the total size of the matrix *p'* to homogenize both spaces.



dinate system to the floor by subtracting the height (H) from this CS to the floor. $|w|$ is the magnitude of the 3D grid and $|W|$ is the magnitude of the continuous workspace. This give us the correlation 2.28. Who translate world positions to the discrete space where we can check if particles are occupying this cell or if we can place a new particle at this place. As we round the positions in the discrete space, when we move the particles back to the continuous space the particle will be centered on the position of the grid, not in its original position, this helps us to arrange all the new particles introduced in the environment evenly. The correlation to bring back the particles to the world space is the same equation 2.28 solved for Q, this is shown in equation 2.29.

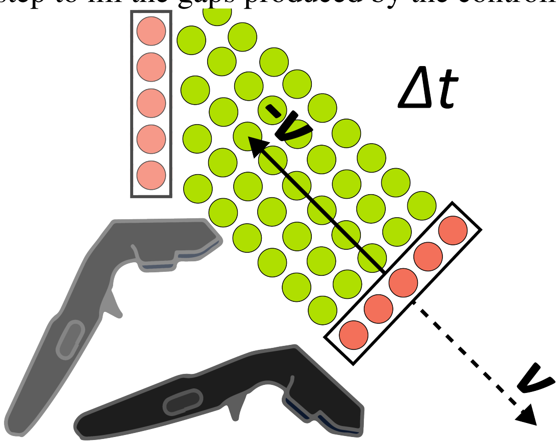
$$Q = (p - c)\frac{|W|}{|w|} + H \tag{2.29}$$

Particle Interpolation

We use basic forms like cubes, spheres, and extension of these as brushes. Figure 2.15 shows the main shapes used to introduce particles to the environment. Starting from a single particle we create primitives by arranging voxels, and with those primitives users can create different forms in the scene. When a brush moves over a volume into space, the positions of the particles on the brush are translated to the discrete space and just after checking the availability of space, particles are translated back from the discrete space to the continuous space. Figure 2.12 shows how particles are introduced to the environment. The particles are introduced according to the control movement in a given direction while pressing the trigger using the shapes mentioned above.

As the hand movements when drawing tend to be faster than the simulation time, i.e., the distance traveled by the controller within a single timestep is bigger than a voxel

Figure 2.17: Particle placement: Particles are placed along the opposite direction of the velocity in each timestep to fill the gaps produced by the controllers travels.



size. This creates gaps between trajectories and non-connected curves in the space; The brushes are used as extrusion planes at each timestep to avoid gaps between positions when the controller moves at high velocities; we have the main shape, and we project that shape into space. To do this, we need a distance in the opposite direction to the movement of the controller. We calculate this distance each timestep using the velocity of the controller, having the velocity magnitude and the timestep we can resolve the length of the movement from the last timestep (Equation 2.30), and from the velocity direction we can infer the direction of the extrusion (Equation 2.31). Figure 2.17 clarifies how this is made, as the velocity direction is in the same of the direction of movement, it is necessary to invert this direction because particles must be added in the already traveled path (the dashed line in Figure 2.17 represents the actual velocity direction, the solid line is the used direction).

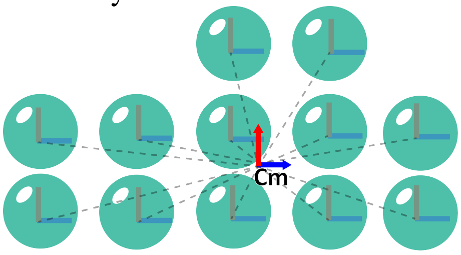
$$i_{max} = Round(\frac{||V||\Delta t}{d_{particle}}) \tag{2.30}$$

As explained above, $||V||\Delta t$ gives us the distance traveled by the controller; we divide this distance by the diameter of a particle to get the i_{max} maximum number of particles we could introduce in that path. Then in Equation 2.31 we proceed to calculate the new positions in the opposite direction of the movement.

$$P'(i) = P - i\frac{V}{||V||}d_{particle} \qquad \forall i \in [0, i_{max}] \tag{2.31}$$

At the end of this stage, we have just position information, but particles are not dynamic yet. We solve this problem in the next section when we talk about particle management.

Figure 2.18: Solid body conformation: Center of mass is calculated and then the differences between each particle and the new center of mass (Cm). Transformations are applied to each particle on the body based on the distances to CM



Particle Management

Having the particles in the space is enough to draw, and create a wide variety of artworks, but not enough to create animations, in this step we make particles behave according to their material properties. The first action we take after the voxelization is to initialize the particles: the phase is chosen according to the user input (Solid, Soft body, Cloth, Fluid). Each type of body has different requirements to interact in the scene, the constraints for each kind of body are created as follows:

Solids: The initialization of a solid body is done first, calculating the center of mass of the entire body by adding all the positions of the particles and then dividing by the number of particles, having the center of mass, the distance between each particle and the center of mass is calculated. The transformations made to a rigid body is introduced directly in its center of mass and then applied to each particle using the relative positions (Figure 2.18). This method is better know as Shape Matching (MÜLLER et al., 2005a)).

Soft bodies: Soft bodies follow a similar path, with the main difference in the number of mass centers. In this case, not just one but several CMs are calculated from cluster of particles. Given a radius R and a minimum distance between clusters dC, some clusters

Figure 2.19: Soft bodies conformation: Several coordinate systems are created based on two properties: a radius r and a minimum distance between clusters dC. Each cluster has a central CM, and the particles inside have a relative position to the cluster(s). Soft bodies behaves as articulated solid bodies

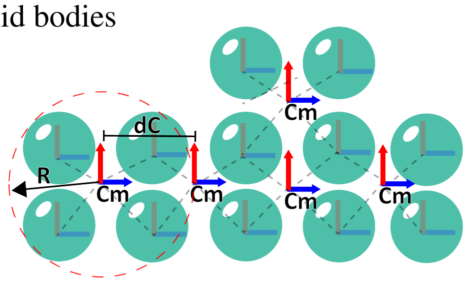
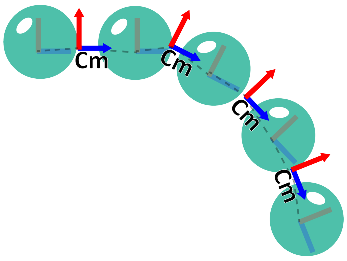


Figure 2.20: Rods and clothes can be created using specific brushes in combination with the soft body option. The behavior of these bodies is similar to the soft bodies simulation.



of particles are created, this results on a structure of articulated bodies inside a total body. A given particle can belong to more than one cluster. As in solid bodies, every particle belonging to a cluster has a relative position to the CM of the cluster (MACKLIN et al., 2014)). Figure 2.19. Shows how the coordinate system is created. A brief look to Figure 2.18 and 2.19 shows the main differences between structures.

The creation of clothes and rods is conditioned by the brush used. As shown in Figure 2.15, some geometries are ideal to create this kind of objects; If a user creates a soft body using the linear brush with a size n, the resulting object will have a behavior of Cloth. If instead of the linear brush, the user decides to create a long rod with a composition of soft body, the behavior will be as shown in Figure 2.20, and will get a rod simulation.

Collision Handling

As our model is built over Flex (by Nvidia), we based the collision detection on the *Flex* Model so, our system detects only particle-particle but not particle-mesh collision. That limitation forces us to model everything in the scene with particles, including the tools (controllers). To handle solid-fluid collisions we modify the radius of solid particles constrained to $R_{Solid} < R_{Fluid}$, using this we prevent possible leaks of fluid particles through thin groups of solid particles. Controller-objects collisions are managed differently: First, the controller mesh is filled with zero mass particles, this to avoid gravity effects on the interaction particles. Collisions particle-particle remains as usual, but the dynamic of those particles is given by the external tracking and not by the simulation dynamics. The collision system remains stable under normal conditions, but there is a maximum velocity where the particles would pass through an object because of the simulation timestep, as this value is considerably high, it is not an issue.

Rendering

The rendering is divided in two parts: Fluids and Rigid-Soft bodies.

Fluid: Fluid rendering was done using Anisotropic Kernels because this is the standard system used by Flex. This approach renders the objects based on the neighboring particles, namely performing Principal Component Analysis (PCA) over the neighbors. More information about this method could be found in the Yu & Turk paper published in 2013 (YU; TURK, 2013).

Rigid and Soft bodies: As our model uses voxelization to arrange particles through space, marching cubes (MC) algorithm (LORENSEN; CLINE, 1987) is a straightforward way to get a conceptual visualization of the objects. During the simulation, we know from the neighborhood search, which particles are on the surface of the object, so we apply MC only on outer particles, reducing computational cost.

2.3.4 Results

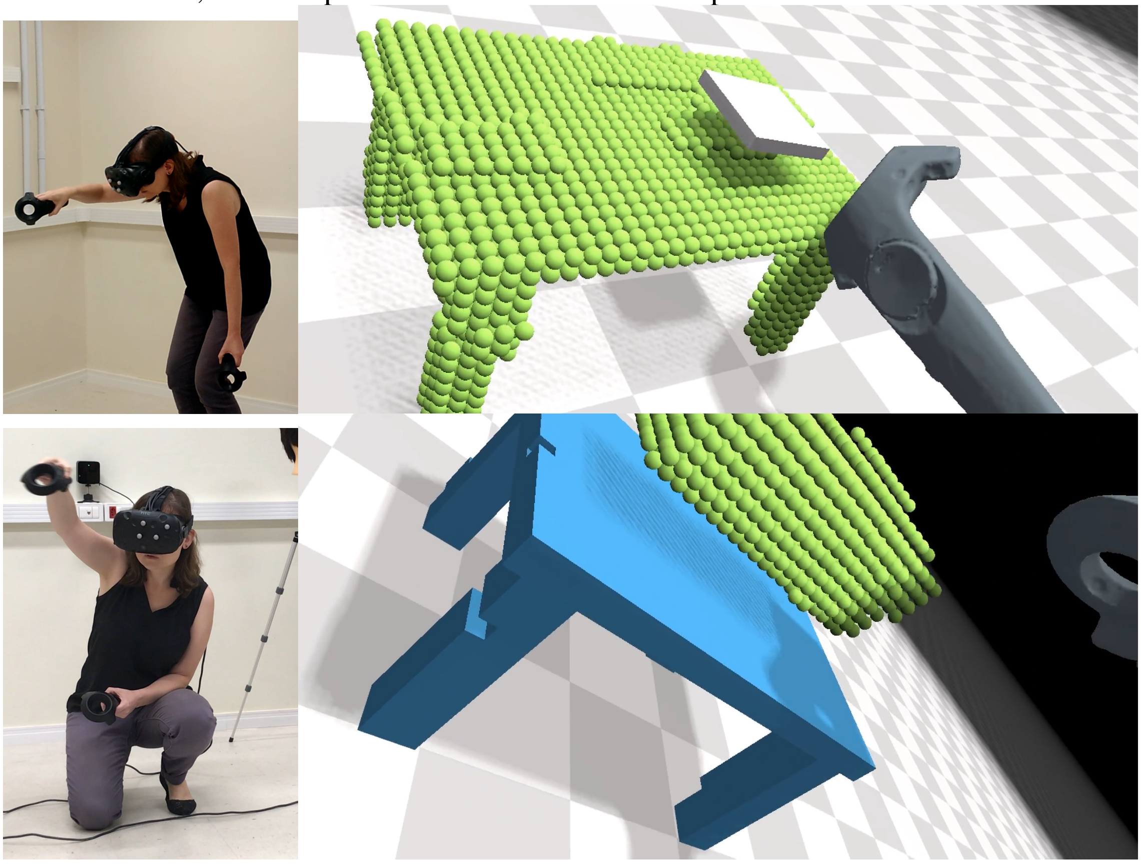
The sketching system was implemented and tested in a Dell workstation with an i7 3.2Ghz Processor, 16Gb Ram, and NVIDIA GeForce GTX 1070 Graphic Card, running Windows 10 and CUDA 9.2. All images and videos used in this section were generated using HTC Vive headset and controllers for visualization and interaction (see Figure 2.21). However, the sketching system is also compatible (and indeed was tested) with Oculus Rift headset and touch controllers. We also successfully tested Leap Motion for interaction.

Performance

Regarding performance, we got average timesteps between 15ms and 33ms; the late was reached when the amount of particles raises over 100k. In this case, the rendering is not fluid anymore. The system represents the workspace in a grid of 151x151x151, where each cell potentially corresponds to one particle. As occupied cells are saved in memory and performance is mandatory, we restricted the max number of particles to 20% of the grid, or 688,590 particles.

To run the scene of the Figure 2.11 a total of 120k particles with a 33ms for frame

Figure 2.21: User creating objects as part of the drawing stage. Upper image: User is creating the table and flattening the surface by erasing particles. Lower image: Having created the table, our user proceeds to create a water recipient on the table.



was used, and the scene of the Figure 2.23, a total of 80k particles with a 28ms for frame was used. There were 8 iterations for the run simulation and with a timestep of 16.6ms.

Expressivity

We invited two subjects to informally test the immersive sketching application to create some sketches using the tools presented in Section 2.3.3. Both had no previous experience with VR. The test was divided into three stages and artists were free to spend as many time as they wanted in each stage:

1. Training stage: The system was introduced to the participants, letting them become familiar with the controls, the virtual tools, and the virtual environment. The system was presented as a tool for 3D sketching in virtual environments, but capable to simulate physic behaviors. So, they first need to choose a material and a shape; Here the brush was introduced to them. We shown them how to change between shapes, sizes, and materials. To conclude this stage, we requested them to do it themselves drawing a free sketch.

Figure 2.22: User sketching objects as part of the expression stage. Upper image: User is building a kind of tree. Lower image: User is interacting with his sketches by touching it.

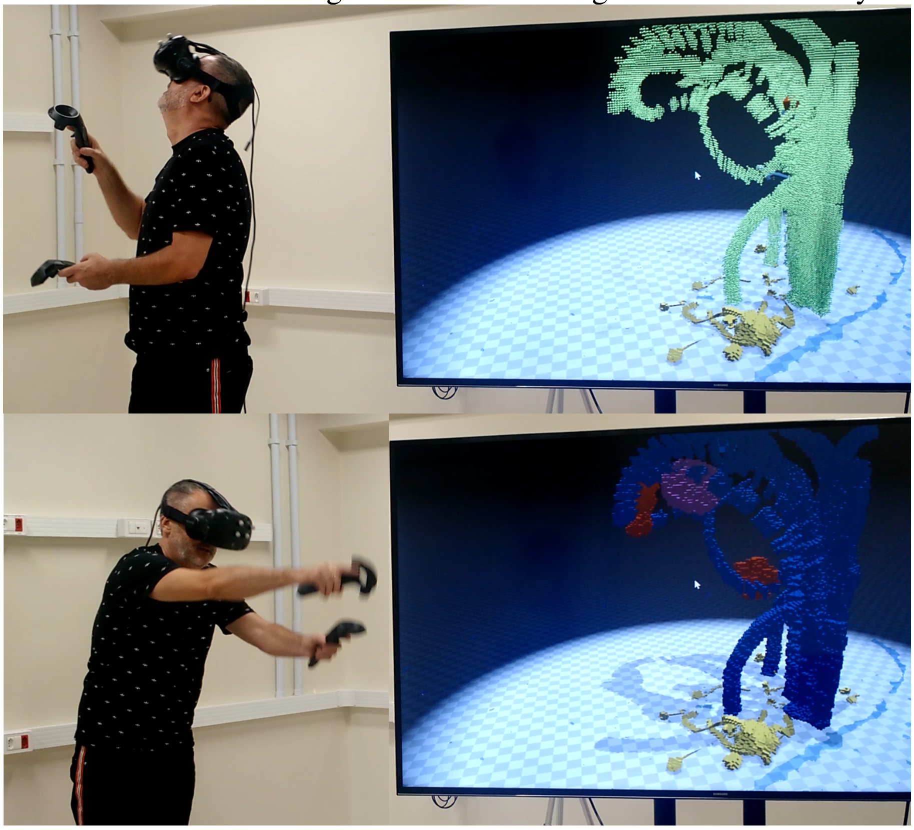


Figure 2.23: In these sequence images we show how our tool allows us to use physics during simulation. This figure shows a pail, a bucket and a rope that joins the pail and the bucket, during the simulation we see how the bucket that has more weight lifts the pail, then we create a container of water that will fill the pail and then match the weights.

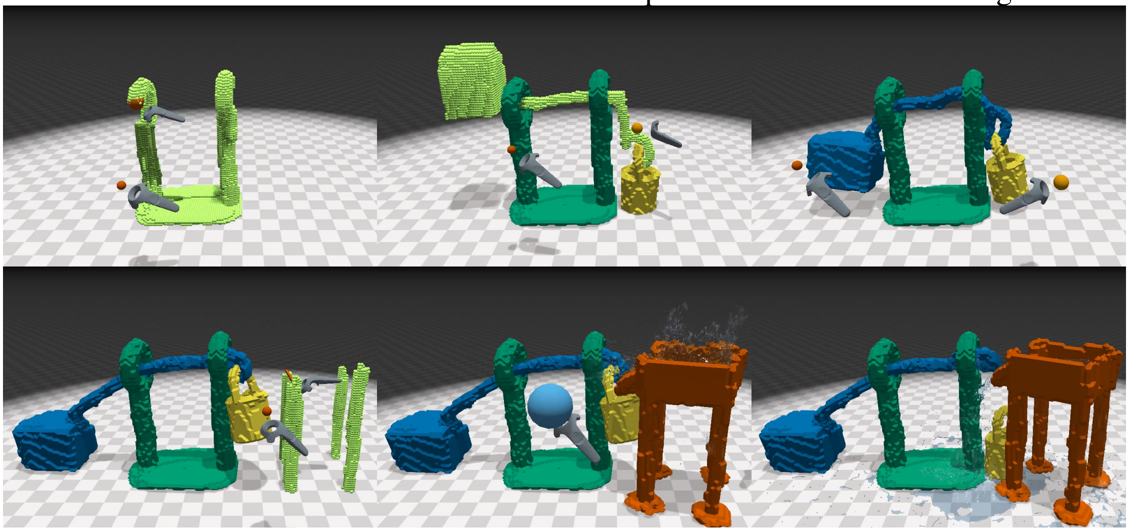
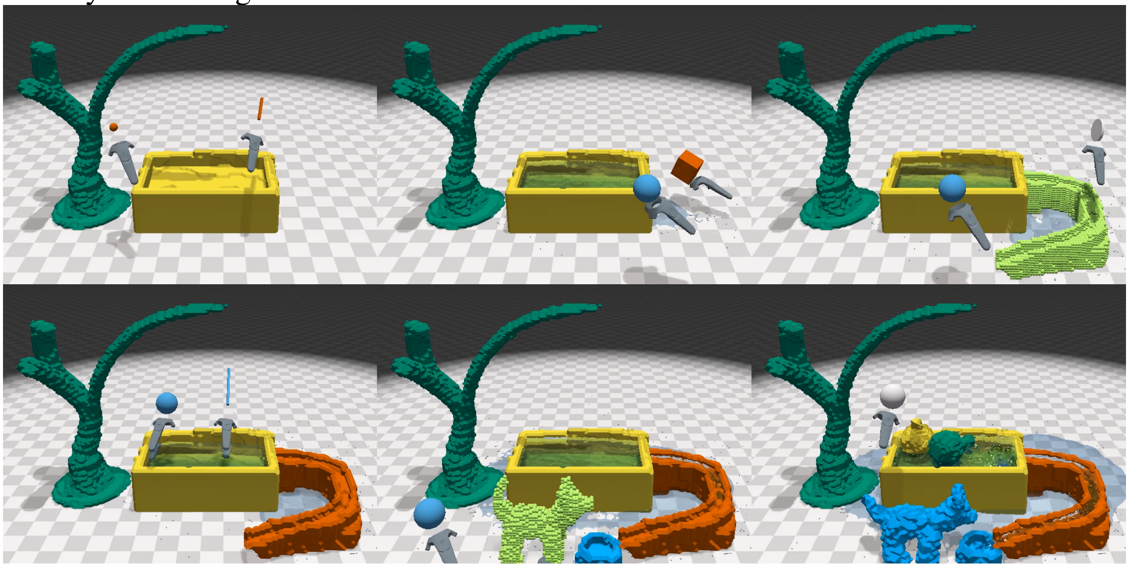


Figure 2.24: In this example we placed a container with a small hole (not visible) in the scene, this reveals the fluid behavior: as long as the water leaks, it flows through the path, to finally fill our dog's bowl.



2. Drawing stage: The artists were asked to draw a table and a container to be filled with water. Then, users were requested to interact with the objects they previously created.
3. Expression stage: The rtists were asked to sketch something original. For this stage, the gravity was set to zero. The sketched objects still reacts according to the materials they are made, but can freely float in the air, what can be seen as an interesting feature for artistics expression.

The experiments took aproximately 40 minutes, where 15 min were spend for training and 25 min for executing the given tasks. Precisely, the first artist spent 40 minutes on the application while the second one spent 35 minutes. None of them reported any sympthon of cyberseekness, and both were very comfortable in the virtual environment.

The artists executed a lot of gestures with their arms, also walking and jumping in the real world. During all the process, they changed their positions – walking and crouching – looking for a better point of view to continue their sketches. An interesting behavior we perceived in both was that they moved in such a way to avoid to collide or even to cross the virtual objects. From observing this behavior and as the result of an informal post-test interview, we can conclude they experienced a high sense of presence in the virtual environment.

Subjects also highlighted the comfort of working with physical objects referring to the soft bodies and water behavior. A drawback reported by one artist was the lack of a strong haptic feedback. Currently, our model only conveys vibrotactile feedback by means of the VIVE controllers when the controller strikes an object in the scene. In the expression stage, the subjects reported an improvement in the experience due to zero gravity. Even though this behavior is non-physical it let them draw objects without building supports.

Another comment of the users is that the best strategy to sketch is using the dominant hand to create particles and the other hand to do different actions, like erasing, for instance. As a suggestion, they highlighted the potential of mixing different materials into a single object, for example, to allows the creation of a solid object with a soft part. They also mentioned that the behavior of the objects is credible, but the rendering of solid and soft objects must be improved. Finally, they also stated that the choice of the color and texture is relevant in the artistic process. However, despite these suggestions for future improvements, they were very excited to keep using the sketching system to create new scenarios and artistic installations.

Figures 2.21 and 2.22 show some frames of the users sketching their artistic installations.

3 TOUCHING THE (ARTIFICIAL) WORLD

The most noticeable presence-breaking factor when immersed in an artificial world is the lack of tactile feedback. In some contexts, not touchable means not existing: Virtual environments, Virtual objects in Augmented reality, even cine directors use the absence of touch to clarify that a given object, subject, or environment is virtual and not real. However, supplying the missing haptic sensations in the artificial world is not an easy task; tactile perception is one of our most accurate senses, and tactile receptors cover us through all our skin (Hale; Stanney, 2004). Moreover, the resolution of those receptors is notably high (BISWAS; VISELL, 2019). There are plenty of approaches to stimulate the mentioned above receptors, as Force feedback, which stimulates several receptors as the muscles, Meissner Corpuscles, Ruffini endings, or vibrotactile devices which elicitates mostly the Pacinian Corpuscles.

Among the full range of available tactile sensations; the study of Stiffness/Softness and Roughness/Smoothness is especially relevant to the HCI community since it directly impacts the object perception. Most of the approaches used to convey stiffness perception are grounded (i.e., using force feedback robots), limiting the available workspace to the device workspace. Additionally, those devices are expensive and out of the budget of most of the users. Furthermore, for roughness displays, it is common to use vibrotactile actuators, constraining its applications just to roughness actuation and not being extensible to stiffness applications. Also, In the latter years, developments and commercialization made on mid-air haptics enabled users to get a haptic interface capable of rendering a considerable amount of shapes and intensities. At the same time, these kinds of ultrasonic arrays come with the significant constraint of grounded feedback: The workspace is constrained to the volume comprised from 15cm to 50cm above the display and a conic-like shape around the display.

In this chapter, we explore two ways to stimulate tactile sensory inputs: 1. Through ultrasound mid-air haptics and 2. Through a wearable finger-mounted device. First, we introduce the concept of ULTRAMOTION, which stands for Ultrasound in Motion, our approach technical-mechanical and aims to expand the capabilities of ultrasound-based devices for haptics. Complementarily, in the second part of this chapter, we present an user-study, evaluating the impact of a wearable device capable of conveying both roughness/smoothness and stiffness/softness with the same set of actuators. Moreover, we focus on softness perception by setting an initial charge on the finger-mounted device and then releasing when touching an object. For the roughness setup, we apply a shear movement to the finger skin surface based on the fingertip velocity.

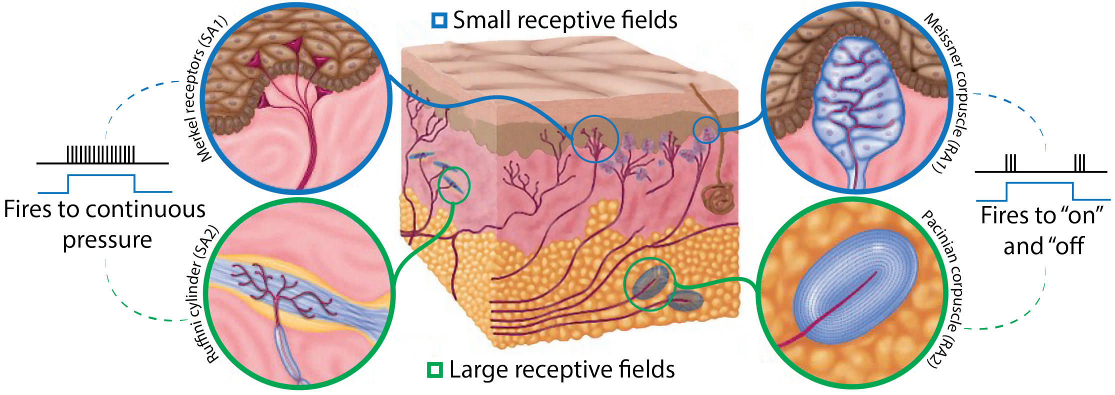
3.1 Background

Humans are provided of nervous receptor in almost all the body, distributed through the skin (which is the human most massive organ). The understanding of how these nervous receptors or mechanoreceptors work is fundamental to design any tactile display; in this section, we present a set of elemental concepts in the understanding of the human haptic sensing.

3.1.1 Cutaneous system

The tactile sensations on the fingers and the hand, could be seen as the sum of the information obtained from the four main mechanoreceptors, comprised by Slowly adapting type 1 (SA1) receptors or Merkel cells, Slowly adapting type 2 (SA2) receptors, also called Ruffini corpuscles (or cylinders), rapidly adapting receptors type 1 (RA1), widely known as Meissner corpuscles and finally rapidly adapting receptors type 2 (RA2 or PC) or Pacinian corpuscles (see Figure 3.1).

Figure 3.1: Cross-section of the skin structure (adapted from (GOLDSTEIN; BROCKMOLE, 2016))



Those receptors have different receptive fields (Small or Large); the cutaneous receptive field is the area of influence of the neuron's firing when the receptor is stimulated. This means that both, Merkel receptors (SA1) and Meissner corpuscles (RA1) have a small receptive field because they are in the surface of the skin, while the Ruffini cylinders and Pacinian corpuscles are located in a deeper position, which means a larger receptive field.

The receptors also can be classified by its behavior when stimulated. Let us illustrate this with Figure 3.1. The blue line shows the response to a given pressure stimulus presented and then remove. The black bars are the response of the receptors, Slowly adapting (SA) receptors fires continuously while the stimuli are present; Merkel receptors and Ruffini corpuscles share this behavior. Merkel receptors are especially useful for detecting textures, and small details and Ruffini corpuscles are known for detecting stretching of the skin. In contrast, Meissner and Pacinian corpuscles fire when there is a change on the stimuli, i.e., the stimuli appear. The later type of receptors are called rapidly adapting (RA) because of this behavior and are responsible for vibration perception, texture discrimination in active touch (RA2 Pacinian corpuscle) and motion across the skin (RA1 Meissner corpuscle).

3.1.2 Tactile thresholds

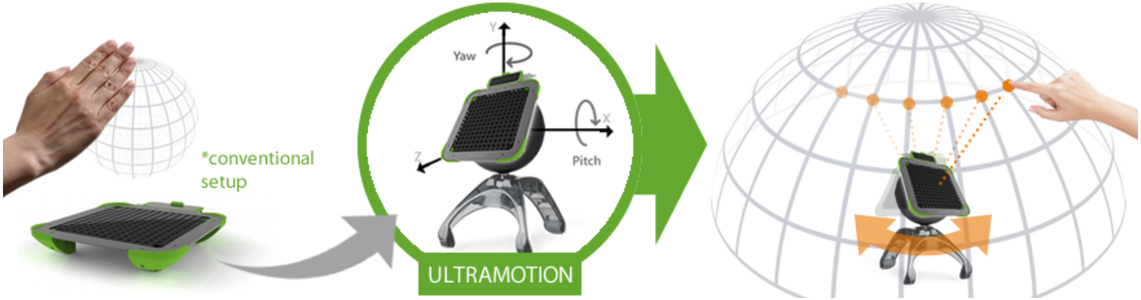
Merkel cells densely populate the skin, in a rate of about 100 per cm^2 (JOHNSON, 2001). Their spatial resolution is 0.5mm, with a receptive field between 2-3mm. Also, SA1 receptors respond to skin indentation of 1500μm minimum. For RA receptors, this value starts at 100μm and saturates with values over 100-400μm. Merkel cells are notably better in dynamic than in static scanning, at least ten times (JOHNSON; YOSHIOKA; VEGA-BERMUDEZ, 2000). Their frequency range is 0-100Hz.

Meissner corpuscles are distributed even more densely through the skin, at a rate of 150 per cm^2. These receptors are sensible over all their receptive field, so their spatial resolution and receptive field size are about 3-5mm in diameter. RA1 are about four times more sensitive to skin deformation than SA1. Their frequency range is 10-300Hz.

Pacinian corpuscles Are distributed throughout the hand, with an amount of 800 in the palm plus 350 per finger. RA2 receptors respond to 10nm of skin motion (amplitude) or less at 200Hz (frequency) (BRISBEN; HSIAO; JOHNSON, 1999), however other authors registered a range from 20Hz to 800Hz. Their receptive field may comprise the entire hand (JOHNSON, 2001). PC corpuscles are particularly sensitives. As a result, these receptors are protected against strong, low-frequency forces. Similarly, a protective task is made by the lamellae, which works as a high-pass filter.

Ruffini corpuscles populate the less densely skin than the receptors mentioned above.

Figure 3.2: ULTRAMOTION moves the Ultrahaptics device to increase its actuation area.



Their receptive field is about five times larger than SA1 and RA receptors, in contrast, they are six times less sensitive to skin indentation but 2-4 times more sensitive to skin stretch. Their frequency range is 0-300Hz.

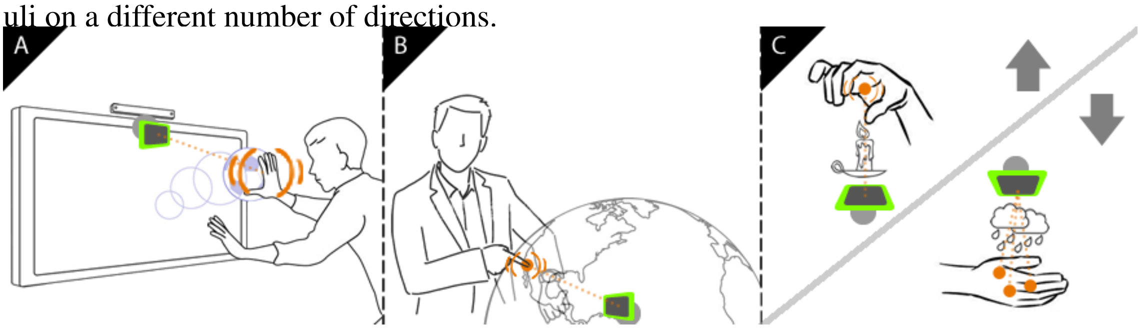
3.2 Touching objects in the air: Enhancing workspace for ultrasound based haptics

This section is based on the following demo:
Steeven Villa Salazar, Victor A. Oliveira, Luciana Nedel, Anderson Maciel.
ULTRAMOTION: Enhanced workspace for ultrasound based haptics. in *2018, EUROHAPTICS Conference*

The volume of actuation of any haptic robot/transducer is fundamental to determine the capabilities of the device. A small mid-air haptic device such as the Ultrahaptics is enough to provide a multisensory experience in most applications, but still, it provides a small working area. We propose a device to enhance the Ultrahaptics capabilities by moving the ultrasound array according to the system and the user needs. Thus, the so-called ULTRAMOTION would provide a more significant working area and new possibilities for applications and interaction approaches. The ULTRAMOTION makes the Ultrahaptics more smart and dynamic. In our demo, we will demonstrate with a goalkeeper game how fun and easy it is to interact with ULTRAMOTION, and how it can expand mid-air haptic device capabilities.

In our project, Ultrahaptics evolves in ULTRAMOTION, a mid-air haptic device that can orbit a point to provide a more significant actuation area (see Figure 3.2). Our proposal gives two degrees of freedom to a device initially thought to work in a fixed place. With our upgrade, the users and developers could enjoy a new horizon of pos-

Figure 3.3: Example of different applications where the ULTRAMOTION device. (A) The device can follow the user's hands to provide directional feedback and even in other body parts for attentional redirection; (B) User can interact with big volumes and holographic visualizations; (C) ULTRAMOTION can also allow Ultrahaptics to deliver stimuli on a different number of directions.



sibilities due to paradigm-shift that ULTRAMOTION brings. Our approach focuses on integrating mid-air ultrasound haptics with robotic kinematics. Attaching the actuator matrix to a 2DOF robot allow us to not only convey haptic stimulus in the original space but in the whole space covered by the robot motion. In that way, the user hand can be followed by the actuator matrix, giving a stimulus to the user even when he/she moves the hand.

3.2.1 Related Work

Mid-air haptics as an option to grounded haptics, exoskeletons or awkward body-mounted devices emerged in the last ten years; promising advances were achieved with the commercialization of phased arrays, or air-based devices. An example of this is AiReal (SODHI et al., 2013), an apparatus introduced by Disney research and capable of generating tactile sensations in mid-air. The device has a large workspace, constrained mainly by the power of the pumps. Also, the bigger the distance from the airborne to the skin, the lower the resolution of the stimuli. On the other hand, it produces an audible sound caused by the actuation of the device. However, this limitation is also shared with ultrasound phased arrays and, understandably, it is hard to remove the noise generated by the natural actuation of the device. Likewise, in the same year, Carter et al. (CARTER et al., 2013) introduced Ultrahaptics, a matrix of ultrasound actuators which render focused ultrasound to convey multi-point stimuli in the bare hand. In the following years, exciting applications using ultrasound were presented: Kovacs et al. (KOVÁCS et al., 2015) generated *Holograms* by using HoloVizio while render focused ultrasound stimuli by mean of two phased arrays placed on the corner of a workbench to add *Tangibility* to the virtual objects. In the same year Sand et al. (SAND et al., 2015) published a work attaching the phased-array to an HMD, enabling users to move through the environment while having ubiquitous haptic feedback freely. However, we perform most of the natural interactions with the back of the hand facing us, and consequently, the users using this approach will not feel the stimuli in the palm of their hands. Two years later, Freman et al. (FREEMAN et al., 2017) published an early work exploring the possibility of rendering textures using ultrasound. More recently, Howard et al. (HOWARD et al., 2019) presented more profound research on finding the thresholds of perception using phased arrays; this includes bumps and holes discrimination and different orientation. However, the vast majority of these works hold the same limitation: they are static arrays with a limited workspace dependent on the volume of propagation of the ultrasound waves. This work aims to extend this workspace by integrating the phased array and a grounded robot.

3.2.2 Methods

Stimulation by ultrasound

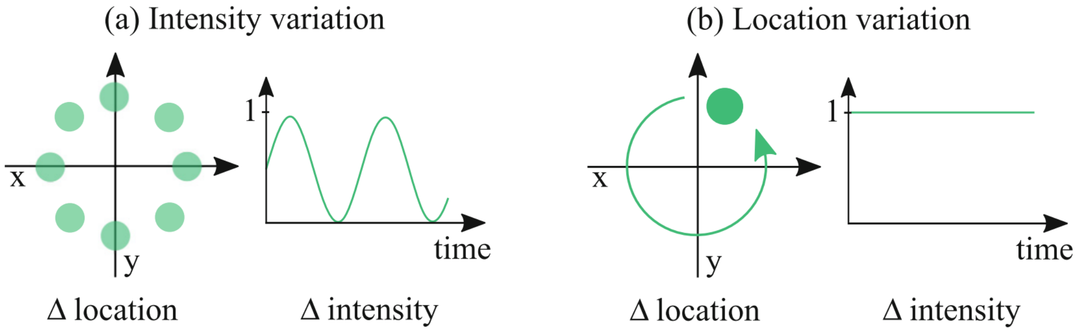
Ultrasound waves move at a frequency higher than 20khz, namely 40khz for Ultrahaptics phased array, and RA receptors have thresholds much lower, i.e., PC receptors have a maximum threshold of 800hz. This issue can be solved relatively easy by downsampling the signal from the range of ultrasound to a hand-sensitive range. Also, the high frequency of the original signal allows modulating different focal points at different frequencies

Phased arrays

Phased arrays are composed of a set of ultrasonic transducers, these transducers (with an approximate pressure of 20 Pascals at a distance of 30cm in the first version (CARTER et al., 2013) but up to 1125 Pa in the version used in this work, which was the same used by Howard et al. (HOWARD et al., 2019)). The stimuli are rendered by calculating an amplitude and a phase delay to create an acoustic field. Next, a table of amplitudes and phase delays are sent the processing unit to render the stimuli using the transducers finally.

Currently, there are two main rendering modes for phased arrays; Amplitude mod-

Figure 3.4: Contrasting Amplitude modulation and Spatiotemporal modulation; The left image displays eight fixed focal points with change on intensity. The right image shows the main concept of spatiotemporal rendering: Move only one focal point at an update rate higher than the lower threshold of the spatiotemporal mechanoreceptors. (Taken from (FRIER et al., 2018))


(a) Intensity variation

(b) Location variation

ulation and Spatiotemporal stimulation. Amplitude modulation (Figure 3.4a) is the most commonly applied in ultrasound-based haptics; it consists on render all the focal points independently and modulates the intensity of the ultrasound waveform to generate a downsampled perceptible waveform. The intensity of the waveform affects the perceived stimuli. An essential drawback of this technique is the fact that the more focal points rendered, the weaker the intensity of each focal point. On the other hand, Spatiotemporal rendering (FRIER et al., 2018) (Figure 3.4a) deals with the problem by maintaining the maximum intensity in a single focal point and moving the point along a given trajectory. The trick is to update the single focal point at a velocity higher than the spatial resolution of the mechanoreceptors responsible for this task; this threshold ranges from 2ms to 40ms (LOOMIS, 1981). Therefore, if this update rate is achieved, the sensation will be closer to a single tactile pattern than a succession of points.

3.2.3 Hardware projecting

Our setup consists of an Ultrahaptics phased array, a haptic robot Geomagic touch a.k.a Omni Phantom, and a computer responsible for all the calculations. The idea of moving the array using a small robot brings some issues related mainly to weight; In nominal position (this is when arms are orthogonal), the device supports up to 3.3 Newtons. That position is the most optimal configuration since it is the one with lower standby torque. It means that the Ultrahaptics weight should be less than 336 grams to be handled by the robot. From Phantom's datasheet, we have $P_{fmax} = 3.3N$ at an orthogonal position, and the torque around the holder is given by:

$$T = (a \cdot cos(\alpha) + b \cdot cos(\beta)) \tag{3.1}$$

We also have that the only orthogonal position of b is when $\beta = -90$, thus we have that maximum torque is $T_{max} = a \cdot P_{fmax}$. Since this torque constrains us, an auxiliar apparatus to handle the array is fundamental. We Designed an apparatus to hold the Ultrahaptics while the robot only actuates on a lever, responsible for transferring the rotations mechanically (keeping our requirement of providing two degrees of freedom). The first design consisted in place the Ultrahaptics at the end of a lever attached to the Phantom (see Figure 3.5 (Left)). The torque required to hold up the Ultrahaptics in this configuration can be easily calculated with the momentums of the apparatus:

$$\sum M = U_w Y - P_f X = 0 \quad \text{with} \quad P_f = U_w \left(Y/X\right) \tag{3.2}$$

Where U_w is the Ultrahaptics weight, and P_f is the force used to hold the device. Here the problem of weight becomes a mechanic problem, where we can manage to move different weights with the same force just adjusting the relation between X and Y. Our main constraint is P_{fmax}, so placing P_{fmax} in the equation and solving for Y/X we obtain:

$$P_{fmax}/U_w \geq F_s \left(Y/X\right) \tag{3.3}$$

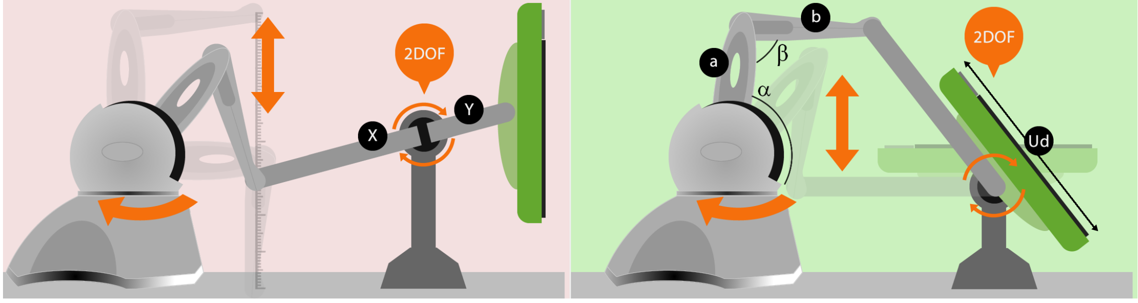
In equation 3.3 we introduced F_s, which is the security factor, in this kind of applications typically $F_s = 2$.

Since our objective is to minimize the right side of equation 3.3, we should make Y less than X. However, the trivial solution for this inequality could be got by doing $Y = 0$ that leads us to the second apparatus (see Figure 3.5 (right)). In this case, all the weight relays the apparatus, excluding the weight of the lever. Thus, the robot should not need to make any force to hold up the Ultrahaptics device. The only opposition to movement is the inertia of the body; if we treat the device as a flat plate, the moment of inertia is given by:

$$U_I = \frac{1}{12} U_w U_d \tag{3.4}$$

With U_d been the distance between extremities of the device (as shown in Figure 3.5). With these values, we can calculate the maximum angular acceleration (φ) that could

Figure 3.5: (left) A lever attached to a holder to reduce the force needed to raise the Ultrahaptics. With this setup, the device would be heavy for the Phantom to support. Our final setup (right) places the Ultrahaptics on a holder, and use the lever to move the holder. This time, the weight of the Ultrahaptics will be supported mostly by the holder, and we still are going to be able to move it with 2DOF



be reached by the Ultrahaptics using the Phantom:

$$T = U_I \varphi, \quad \text{with} \quad \varphi = 12 \frac{(a \cdot P_{fmax})}{(U_w U_d)} \tag{3.5}$$

Then, the optimal configuration of the apparatus is shown in Figure 3.5 (right).

3.2.4 Hardware construction

The manufacturing process relied on 3d prints and mechanical parts as screws. We printed the interfacing pieces to attach the arm of the Phantom device to the holder supporting the Ultrahaptics. Our design process began modeling a primary container to put inside the Ultrahaptics. We iterate several designs to achieve the right size and reinforce the critical areas of the holder. Figure 3.6 Left shows the main stages of the holder design process: The first design was targeted to contain the boundaries of the Ultrahaptics device accurately. In the center, the second print focuses on reducing the quantity of material and improve the stability of the Ultrahaptics. Additionally, in the center of the holder, we placed a region to connect with a conventional camera tripod head. The last image of Figure 3.6 shows the final model with improvements in the tripod head adjusts and four grabs; two laterals and two in the bottom corners.

Figure 3.7 shows the details of the grabs on our final model. The lateral grabs prevent the device from slipping off the holder and support its weight as well (Figure 3.7 left). On the other hand, the bottom support grasps the corner of the Ultrahaptics, giving even more security and stability. Those supports are capable of holding the device even upside down.

Figure 3.6: Different alternatives for the 3D printed piece made for holding the Ultrahaptics device: Left: early prototype used to verify dimensions, Center: Second model with a more robust composition and less material. Right: Third model with lateral supports and grabs in the bottom corners

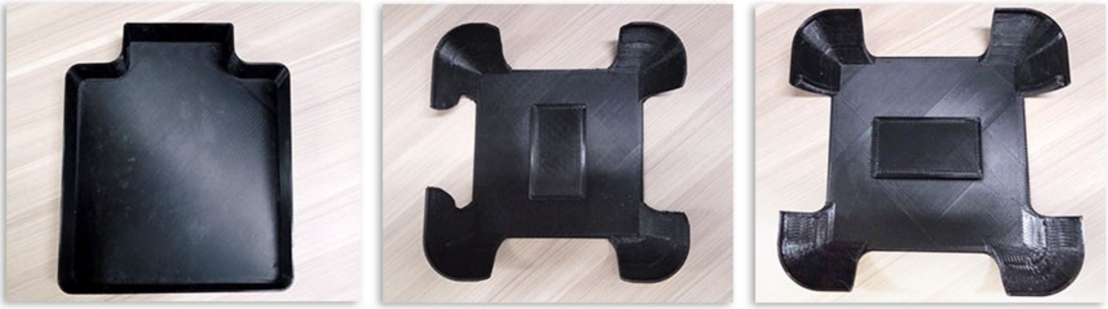
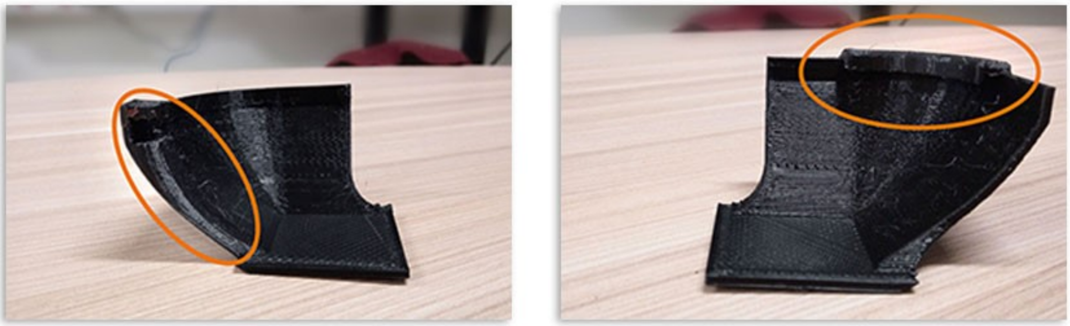


Figure 3.7: Details of the security grabs placed in the holder



After assembling the whole setup, we confirmed that the ULTRAHAPTICS could be controlled to face the directions and that the holder would hold the device from falling. A final cover for ULTRAHAPTICS was designed to support the device even when the ULTRAHAPTICS gets elevated at $90°$ from the table. We also added extra weight to the holder to keep the device from drifting during its manipulation. The figure fig:UH9 Left, Highlights the joint and shows the connection to the holder, the base, and the robot, the second image shows we used weights to guarantee stability.

Figure 3.8: The manipulations are done through the level (left). An extra weight keep the ULTRAMOTION from drifting (right) during manipulation.

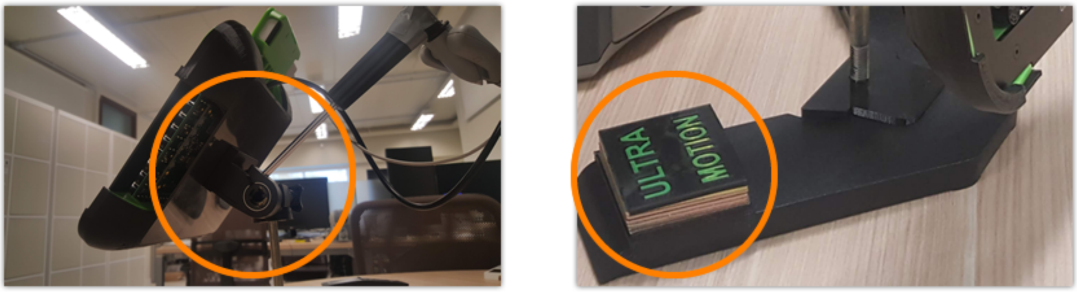
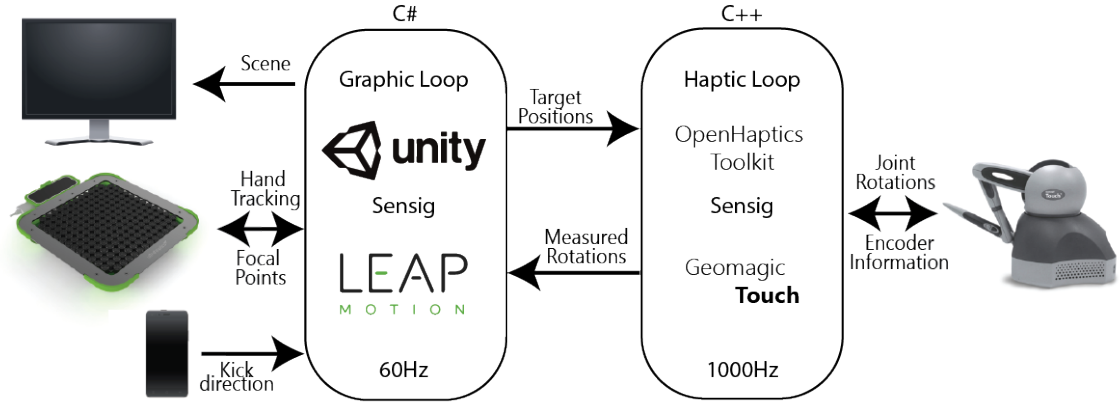


Figure 3.9: Overview of the architecture of our application



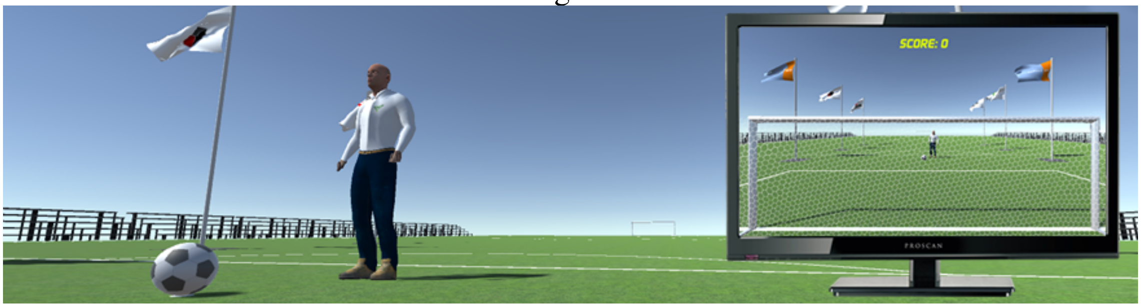
3.2.5 Software

Figure 3.9 shows the structure of our application; we based our code in OpenHaptics toolkit under C++ to control the Omni Touch device. Additionally, we built the graphic scene using Unity 3D. We created two loops to control the system based on the update rate of the devices/Algorithms. Graphics, hand tracking, coordinate transformation, focal point rendering were implemented to work at 60Hz, which is a successful framerate for graphics. Also, the leap motion works at this frequency. Furthermore, Haptic loop was set to update at 1khz to avoid instabilities in the robot; a closed control loop tries to set the end effector in the target position given by the graphic loop at a lower frequency. This scheme lets us integrate both seamlessly. The data shared between loops is Directional force, target position, Rotational force in each joint.

Due to the movement of ULTRAHAPTICS and the sensing device (Leap Motion) the range of the leap motion would be minimal as it cannot be fixed in the workspace due to the working volume of ULTRAHAPTICS which would be bigger than the sensed working volume, or, in other words, the workspace of the infrared cameras of the leap motion. Then, a new coordinate system should be defined, using the robot end-effector position (here will be the Leap Motion) plus the position measured by the Leap Motion (to get the absolute coordinates). The new workspace is obtained by applying the transformation matrix of the end effector to the Leap Motion coordinate system.

This approach is hand-centric, so the target positions sent from the graphic loop obeys the inverse kinematics of the system to place the phased array facing the palm. The optimization parameter is the distance from the hand to the center of the coordinates system of the ultrahaptics device. When no hand is detected, the apparatus stays in the

Figure 3.10: Our demo game depicts the penalty area with a virtual soccer player ready to kick the ball. The scenario contains some decorative flags with logos of our research group, department, and university, as of the EuroHaptics conference and the Ultrahaptics. The hands of the user are shown inside the goal.



last position. Additionally, as the array is continuously moving, the focal points could be offset from the target position and not hit sometimes in hand. It introduces a risk to the health of the users: the undesired propagation of the modulated waves towards the ear. To avoid this, we modified the rendering Algorithm to add security filters in risky situations, i.e., the hand is moving or waving to fast. Also, if the tracking sensor does not detect the hand, no stimulus is provided

3.2.6 User case: The goalkeeper

We created a virtual scene to show the potential of our application. In that scene, the camera is placed behind the goal facing the avatar of a soccer player already placed at the penalty kick spot with a soccer ball (see Figure 3.12). When the user places one hand over the ULTRAHAPTICS device, it becomes visible to the LeapMotion, the virtual representation of the hand appears inside the goal. A virtual box representing the ULTRAHAPTICS is centered at the goal to show how it moves following the virtual hand.

The demo includes the implementation explained above to bring the user gaming experience. Participants should play as a goalkeeper and to stop the balls coming from the front of them (see Figure 3.11). While he is watching the balls get closer to him, he should block them. If the ball is blocked, haptic feedback is given.

We also developed an Android app showing the penalty area from the perspective of the soccer player (see Figure 3.12). The app is connected to the game by wifi (using TCP IP protocol). Users can touch any point of the goal to make the avatar kick the ball in the same spot. If users touch the ball instead, the avatar will kick the ball randomly.

The avatar's animation was also chosen to reduce the impact of latency of move-

Figure 3.11: Demo setup. The ULTRAMOTION will be implemented with a Phantom Omni device for the prototype. Participants will use it in a goalkeeper game. The ULTRAMOTION will turn to the right or to the left to render mid-air haptics in the hand equivalent to the side the ball was thrown.

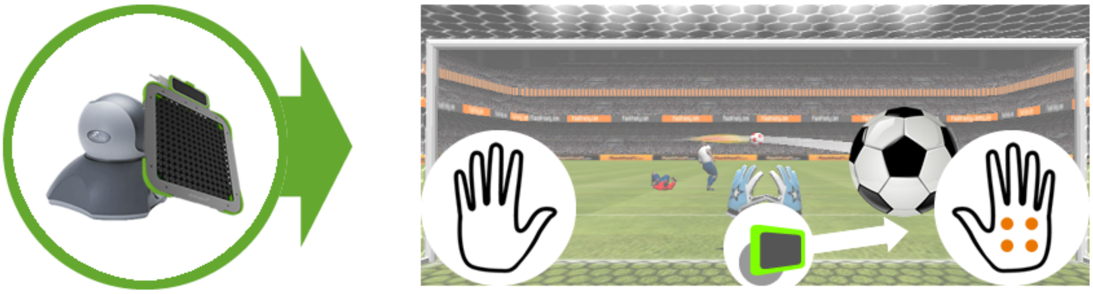
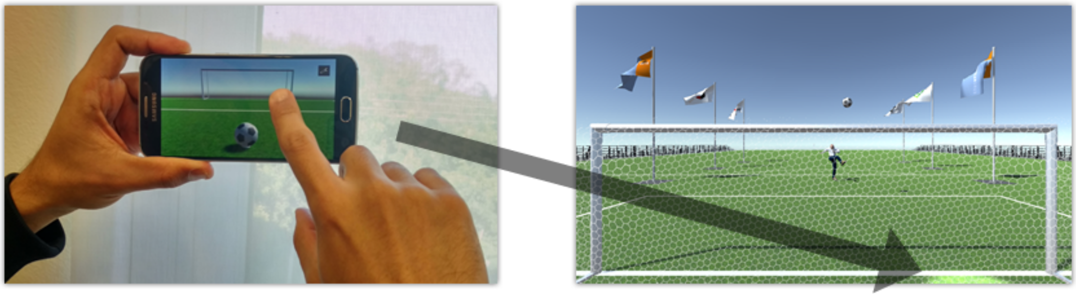


Figure 3.12: Users can see the penalty area and chose where to kick the ball.



ment over the ULTRAMOTION setup. Once the spot was chosen for the ball to be kicked at, the avatar gets ready to kick. That means that the avatar finishes its animation state of idle (3s) and then pass to the animation state of kicking (2.6s). The transition between states gives time for the ULTRAMOTION to follow the user's hand and be positioned at the time to render the tactile feedback.

3.2.7 Discussion

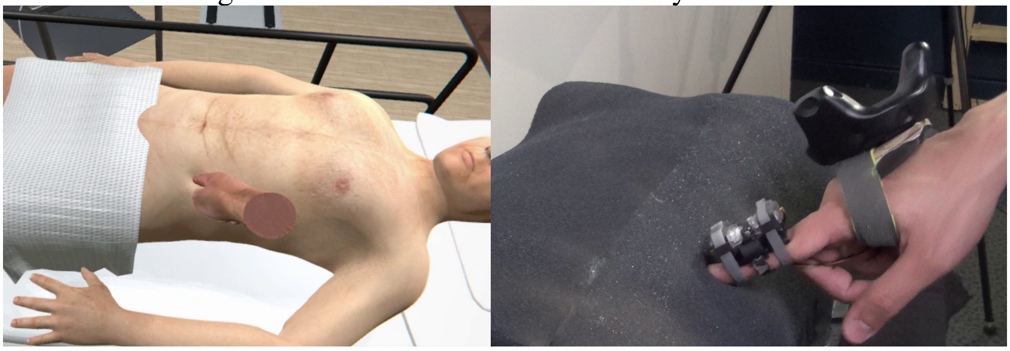
With a moving device, it becomes possible to develop several applications (see Figure 3.3). Possible examples would involve collaborative applications, full-body stimulation, games, among several others. For instance, one can imagine a virtual ping-pong game where two users play with the ULTRAMOTION. In one side a user hits the virtual ball, feeling the hit through mid-air haptics. Then, our device calculates the position and velocity of the ball to move to the other side in a way that the other user can feel the sensation when he/she intercepts the ball. Such interaction would not be possible with a static device though. Besides, the versatility of ULTRAMOTION opens the possibility of taking mid-air stimulation to other body parts. Thus, if placed in front of the user, our application can render haptic sensations on the neck, shoulders, chest, and even in the face (if necessary and safe) in order to provide multi-sensory experiences and even for attentional redirection. Having the capability of conveying stimulus to those body parts allows haptic experience designers HaXD to develop more valuables applications and sensations.

3.3 Feeling Softness: Exploring the use of wearable devices

This section is based on the following publication:
Steeven Villa Salazar, Xavier de Tingy, Anderson Maciel, Maud Marchal, Claudio Pacchierotti. Altering the Softness, Friction, and Shape Perceptionof Tangible Objects in Virtual Reality Using Wearable Haptics. in *For Submission*

Stiffness is one of the most important properties when touching an object, rendering stiffness using force feedback devices was widely explored. While it can be said that robotic stiffness rendering achieved maturity in terms of stability and reliability, wearable stiffness rendering still is in an early stage. Portability of non-vibrotactile haptic devices brings an exciting challenge to the state-of-the-art technology in areas as new materials, actuators, low energy components, light batteries, and miniaturization in general. However, not only the hardware needs to be improved but also the understanding of how humans perceive those stimuli, especially when they are not as strong as in actual object palpation. This work is framed under a set of evaluations and improvements around a specific finger-mounted wearable structure called Hring (Haptic Ring). It was presented by Pacchierotti et al. ((PACCHIEROTTI et al., 2016)) applying a concept previously introduced by (MINAMIZAWA et al., 2007). This wearable model was used previously for controlling extra robotic fingers, (HUSSAIN et al., 2016), to help in the missing grasping capabilities in chronic stroke patients (HUSSAIN et al., 2017), augmented reality (using Hololens) (MELI et al., 2018a), Thermal discrimination (attaching a Peltier cell) (GIOIOSO et al., 2018) and also to augment stiffness of tangible devices (TINGUY et al., 2018). This work aims to continue exploring the range of possibilities which this device can provide; namely, we want to explore the augmentation of softness perception and the possibility of rendering friction/textures.

Figure 3.13: Palpation example using our methodology to render softness in a medical application: When he user is not touching, our device renders a constant pressure, when he/she touches a stiff part, like a rib, it releases in a slow rate, when he/she touches a soft part, it releases in a higher rate to make users feel the body softer.



3.3.1 Related work

Although haptics is present in a significant portion of our daily-life devices (Smartphones, smart bands, game-controllers), most of those products under-use the potential of haptic technologies as a way to communicate information or expand user-experience. In this section, we explore the current efforts made by the haptics community to study haptics within the Stiffness and roughness context, paying particular attention to ungrounded devices.

Stiffness displays

Probably the most popular approach to convey stiffness sensations is the use of robotic devices such as Geomagic touch (TOUCH, 2016) (former phantom Omni) or Omega (DIMENSION, 2013) (by Force dimension). However, these interfaces are sharply limited in the workspace and generally expensive to the everyday user. Several alternatives have been reported in the last years; for example, the usage of exoskeletons shown satisfactory results. Zubrycki and Granosik (ZUBRYCKI; GRANOSIK, 2016) proposed a Glove-Based device provided of jamming tubes; the glove works by filling with air each tube placed in the hand and creating a sensation of stiffness, however, one major drawback is the time taken by the pumps to fill a given tube (around $0.5s$). Moreover, the hydraulic setup makes it troublesome to be taken into account as a feasible option. Later, Maereg et al. (MAEREG et al., 2017) built a wearable vibrotactile prototype based on ERM actuators. Even if this device is notably less cumbersome and improves the wearability; vibrotactile feedback stimulates rapidly adapting receptors as Pacinian or Meissner corpuscles (JOHNSON, 2001), while stiffness is experienced in the real world as stretches and deformations in the fingertip. However, these phenomena are predominantly measured by slowly adapting receptors as Merkel cells or Ruffini corpuscles (GOLDSTEIN; BROCKMOLE, 2016). Similar approaches on sensory substitution using vibrotactile stimulation were presented by Yem and Kajimoto (YEM; KAJIMOTO, 2018) and Sorgini et al. (SORGINI et al., 2018) in 2018, but again, they target rapidly adapting instead slowly adapting receptors. In the same year, Hosseini et al. (HOSSEINI et al., 2018) presented ExoTen-Glove, a haptic glove which uses Twisted string actuation to provide force feedback in grasping tasks. Although the authors claim their design to be lightweight and straightforward because of the absence of gears of other messy mechanisms, the truth is that the size and wearability of the device are not optimized since it has straps across the arm, not just the hand. Similarly, Hinchet et al. (HINCHET et al., 2018) developed DextrES; a promising electric-brake based glove capable of providing force feedback and cutaneous feedback, this glove is considerably smaller than the one presented by Hosseini et al. (HOSSEINI et al., 2018), so its wearability is excellent. However, until the publication of this manuscript, DextrES still operates at voltages over 250V, which is a significant concern to de daily user. Additionally, most of the above-presented approaches do not allow friction stimulation to provide roughness and softness perception.

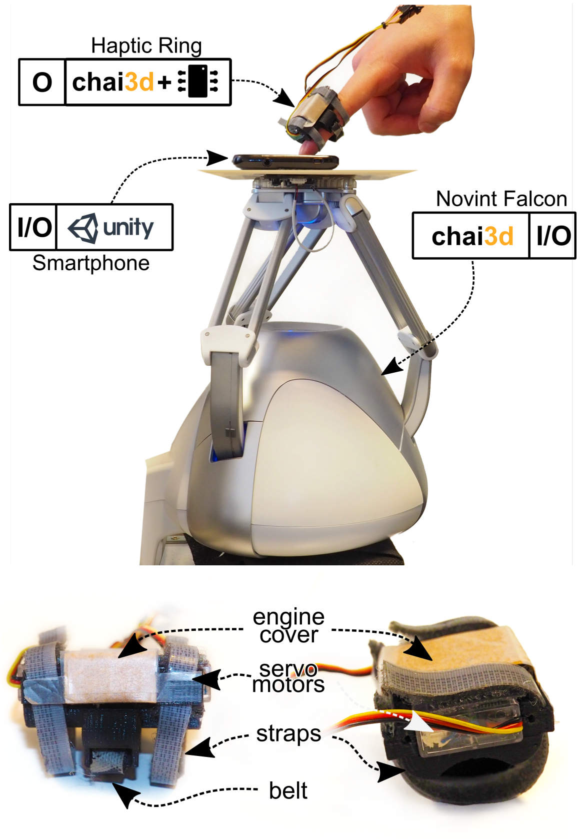
3.3.2 Apparatus

Hardware

We based our setup in two main haptic hardware pieces: A grounded force-feedback robot (Novint Falcon) and a wearable finger-mounted device. The falcon device is a delta-type robot, with a maximum force of 9 N and position resolution of 160 increments per cm, used to measure the position and render a spring force in the z-axis. We configured our render workspace to match with the higher accuracy regions of the robotic device, according to (Martin and Hillier. 2009) (MARTIN; HILLIER, 2009). Additionally, we rotated the Falcon device 90° from its normal usage position (As visible in Figure 3.14). On top of the end effector, we placed a 3D printed platform to hold the input device; in this case, an Android smartphone.

The wearable device (Figure **??**) is closely inspired by the prototype presented by Pacchierotti et al. (PACCHIEROTTI et al., 2016), and recently studied in contexts as augmented reality (MELI et al., 2018b) and tangible interaction (De Tinguy et al.,

Figure 3.14: Main setup: Our setup is composed of: 1. Our wearable device, 2. a Novint Falcon device, 3. An Android smartphone, 4. A laptop resposible of the processing (not visible)



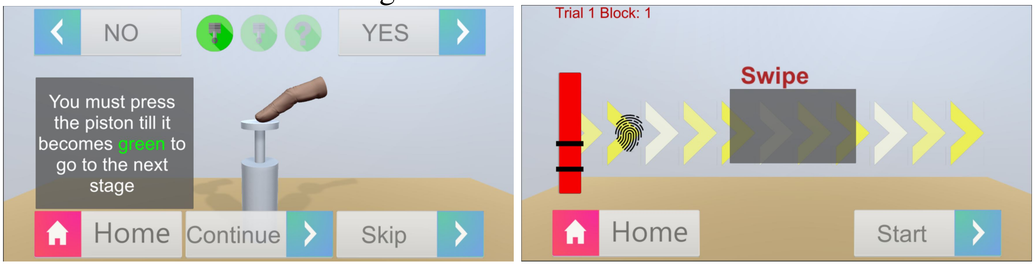
2018). Its body is made of PLA plastic, and it encapsulates two servos HS-5035HD with a nominal force of 7.8 Newtons and a stall torque of 0.8 Kg-cm, Manufactured by Hitec RCD Korea inc. The fabric belt placed in the center is the central actuating part: by pressing/releasing or moving the belt from one side to another, it is possible to stimulate the finger in a variety of ways. The strap is driven by the servomotors, which are both controlled by a Micro Maestro 6-Channel controller.

Finally, the whole system is controlled by a conventional laptop which is connected to the Micro Maestro controller, the smartphone, and the Falcon robot.

Software

We structured our setup to respect the separation of tasks involved in the interaction processing. Our haptic loops (Kinesthetic and tactile) were built on chai3d, so we are able to guarantee a refresh rate of 1000 Hz for the force feedback rendering and approach the maximum refresh velocity of the servo-motors. Similarly, graphic processing, as well as the interaction, were developed using Unity3D. As shown in Figure 3.15, our system

Figure 3.15: User interface



You must press the piston till it becomes green to go to the next stage

was focused in touch interaction in order to be controlled from a smartphone (touch input). Furthermore, we handled the data sharing between applications (Haptics-chai3d UX-Unity3D) using OSC protocol (SCHMEDER; FREED; WESSEL, 2010).

In summary, we have:

Table 3.1: Device list

Device	Input	Output
Smartphone	Touch	UI
Falcon	Position/Force	Force Feedback
Hring		Pressure/Shear Feedback
Laptop		UI

3.3.3 Experimental procedure

Experiment 1. Stiffness

The purpose of this experiment was exploring the plausibility of using the Hring to convey softness sensations in the users. As De Tinguy et al. reported in 2018 (De Tinguy et al., 2018) using a similar device: Stiffness can be augmented by increasing the pressure in the finger as long as the user presses a piston. Having that in mind, we hypothesize that the inverse sensation can be conveyed if the rendered stimuli are inverted. In other words, we can augment softness perception if a standard force is applied in the resting state of the haptic device, and eventually, this force is reduced as long as the user presses a piston.

To evaluate this scenario, we stated two conditions:

- **C1.** The haptic device starts with a pre-charge applied in the middle phalanx of the index finger from the right hand and releases dynamically as long as the user presses the virtual piston.
- **C2.** Constant charge at the middle phalanx.

Participants: Fourteen healthy participants took part in the experiments (12 males, one female, one non-identified) with ages ranging from 18 to 38 years old, all of whom were right-handed. Most of them (57.1%) had previous experience with wearable haptic devices, and 64.3% had previous experience with force feedback.

Protocol: At the start of the experiment, users were informed about the experiment, and the task then asked to sign a consent form (Appendix 10). After, an animated tutorial was presented on the screen (as shown in Figure 3.15). Next, any doubt was clarified, and the actual experiment started. During the experiment, subjects wore headphones playing pink noise. The experiment was conducted in a quiet room, free of distractions.

Participants had to compare between two virtual pistons presented sequentially; Stiffness was rendered by the Falcon robot using a simple spring model $f = -k\Delta z$, being Δz the displacement in the z-axis of the Falcon (See Figure 3.14). To this end, subjects were required to interact with the first piston during two seconds; then a screen informed them to release the first piston and wait to the second one, during this time, the end effector returns to its original position and enables the user to interact with the second piston in the same way. Last, Users were asked to judge, based on their perception, if *piston one* felt **stiffer** than *piston two*. One of the two pistons was rendered as a reference and the other as comparison piston. The total rendered stiffness was rendering using both, the falcon device and the Haptic ring, this is: $k_{total} = k_{Falcon} + k_{Hring}$ with $k_{Hring} = precharge - f$ in **C1** and $k_{Hring} = precharge$ in **C2**. Where k_{Falcon} took two different values per trial; k_{ref} and $k_{variable}$, depending on the piston evaluated. Through a given block of trials, $k_{ref} = 0.1N/mm$ remained unchanged and $k_{variable}$ took seven different values $k_{variable} \in [-92.3\%, -61.5\%, -23\%, +0\%, +23\%, +61.5\% + 92.3\%]$ of the reference piston. The order of k_{ref} and $k_{variable}$ and **C1** ,**C2** was counterbalanced to avoid any order effect. In total, each user was presented with 70 trials: Two conditions (**C1** ,**C2**), Seven values of variable stifness (-92.3%, -61.5%, -23%, +0% ,+23%, +61.5% +92.3%), with five measures per variable value.

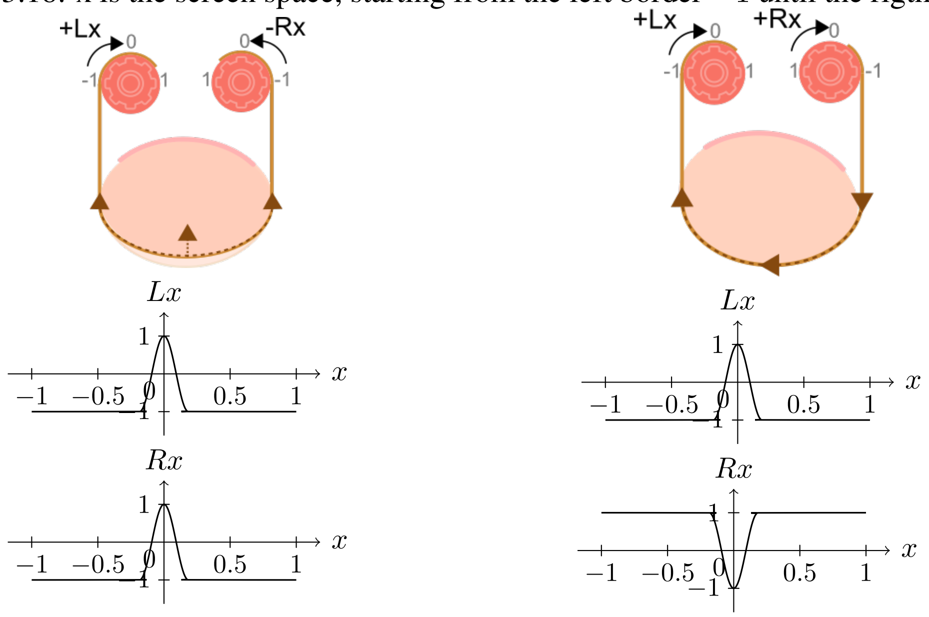
Collected Data: For each trial (two presented pistons plus one question), we measured the subject's answers. The answer corresponds to the perception of the second rendered piston regarding the first one, so, the answer is a "Yes" or "No" value. A final subjective questionnaire was applied at the end of the experience. A typical 7-item Likert scale was used in each question:

- **Q1.** The haptic device on your finger contributed to the perception of stiffness.
- **Q2.** The combination of both cutaneous and kinesthetic sensations contributed to the perception of stiffness.
- **Q3.** The action of the tactile device affects my perception of stiffness
- **Q4.** Practicing improves the association of both cutaneous and kinesthetic sensations
- **Q5.** After the experiment, I felt tired.
- **Q6.** The wearable device in my finger distracted me from the primary task.

Also, the user who wanted to talk about their experience and remarks were shortly interviewed in the post-experiment stage.

Experiment 2. Friction

Figure 3.18: x is the screen space, starting from the left border -1 until the rigth border 1

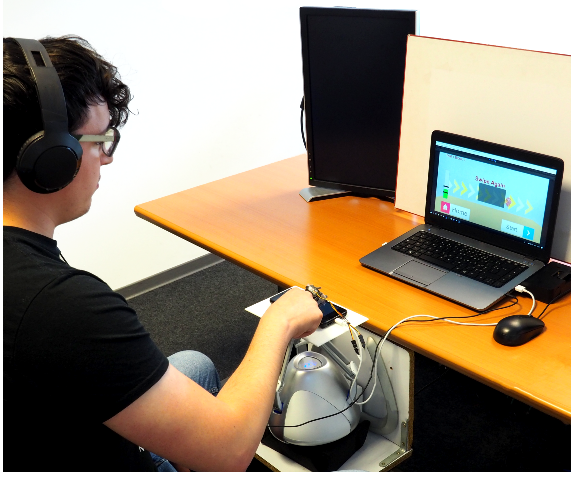


Rendering by pressure

Rendering by shear

Vibrations and movement profoundly influence friction perception; in fact, there is no friction perception without movement because when there is no relative movement, only slowly adapting receptors (as explained above) keep firing signals (FAGIANI et al., 2011). Also, most of the approaches published to render friction in the fingertip uses vibro-tactile actuators (WIERTLEWSKI et al., 2014), (CULBERTSON; KUCHENBECKER, 2016), or encounter-type haptics (NISHIMURA et al., 2014), achieving relevant results. In this experiment, we wanted to study the feasibility of extending the

Figure 3.19: Friction setup: Subjects sats down in front of the setup and performs the swiping looking at the screen.



functionalities of the H-ring by evaluating the friction perception of the users in a sliding task. Previously, similar approaches (PROVANCHER; SYLVESTER, 2009) reported the accomplishment of this objective, even using one servo.

To evaluate this scenario, we stated three conditions:

- **C1.** Rendering by pressure in a 2-AFC design (Figure 3.18 left)
- **C2.** Rendering by shear in a 2-AFC design (Figure 3.18 right)
- **C3.** Rendering by shear in a Yes/No design (Figure 3.18 right)

Although we evaluated all the conditions in the same session, each condition had a variation in the protocol: During Condition **C1** we presented *Bumps* and *Holes* by pressing in the subject's fingertip to render *bumps* and releasing to render *holes*. The question related to this condition was *"What did you feel?"* and the two forced choices were *Bumps* or *Holes*. Each trial consisted of one stimulus and one question. Similarly, Condition **C2** consisted of a stimulus and a question (the same as above), but this time we switched the rendering mode to shear force (the one related to friction). We also presented *Bumps* and *Holes* but using the configuration shown in Figure 3.18 right. This is, *Bumps* as shear in the opposite way of sliding, and *Holes* in the same way of sliding. Finally, in condition **C3**, we changed the protocol for a Yes/No design. Namely, we presented two different stimuli per trial and then asked the following question: *"Is surface 1 slippier than surface 2?"* with two possible answers, *yes* or *no*. The rendering was the same as **C2**, but we presented it in both ways; the same direction and opposite to the swipe direction in the same trial.

Participants: Fourteen healthy participants took part in the experiments (13 males, 1 female) with ages ranging from 20 to 33 years old. One of them was left-handed, the rest were right handed. Half of them had previous experience with wearable haptic devices

Protocol: We asked each subject to read the instructions sheet (Appendix 9) to get familiar with the experiment. After that, any doubts were solved. Next, subjects were asked to fill up the consent form (Appendix 10) where they agree with the conditions of the experiment. Also, we asked participants to fill up a demographic questionnaire before the hands-on tutorial. Then, a training trial was presented in order to check if the participant understood the protocol. At that moment, the actual experiment started: The conditions were presented in a different order based on Latin square to avoid any order impact over the experiment. Before each condition, an animation was presented on screen to clarify what the subject was expected to do. After each condition, users were asked to fill out a questionnaire about the condition experienced. Finally, subjects were asked orally about their opinion of the sensations they experienced. During the experiment, subjects wore headphones playing pink noise. The experiment was conducted in a quiet room, free of distractions.

During the experiment, the swipe direction changed randomly; this value was counterbalanced to have an equal amount of swipes en each direction. The pressure of the fingertip was controlled by asking subjects to keep the feedback bar (on the left of the screen, visible in Figure 3.15 right) within de green range. The force measurements were made using the falcon device. The range of movement of the robot in z axis was also constrained (locked) to not difficult the swiping. Considering the possibility of having unwanted feedback, we covered the mechanical pieces of the haptic ring during this experiment. Having visual feedback of the Hring's actuation could induce bias. Also, subjects were asked to look at the screen while swiping and at smartphone when answering, trying to reduce even more the possible bias.

The stimulus was rendered based on the screen coordinates of the smartphone, Figure 3.18 shows the functions used to control the servomotors resulting from equation 3.6, where $p\!\!\mathbb{Z} \neq 0 \in [-1, 1]$ defines the direction of movement of the servomotor, and a is the width of the bump. In this way, if the user touches a region where $|x| \geq a$ no stimulus is rendered until the swipe reaches the bump or shear area.

$$shear = \begin{cases} p \cdot Cos(\pi \frac{x}{a}) & |x| < a, \\ p \cdot Cos(\pi) & Otherwise \end{cases} \tag{3.6}$$

To render condition **C1**, we sent the same signal to both servomotors in a given direction $p_{1,2} = 1$ or $p_{1,2} = -1$. In contrast, **C2** and **C3** were rendered using opposite signals, this is: $p_1 = 1 \& p_2 = -1$ or $p_1 = -1 \& p_2 = 1$. The final stimuli depend on a and p values as well as the velocity of swiping and pressure applied in the fingertip. Also, the range of actuation of the Haptic ring depended on the size of the finger of each subject, we adapted the pressure to the different sizes by putting spacers between the Haptic ring and the finger, and most importantly, we calibrated the maximum and minimum ranges according to the individual's finger. After all, we put special effort into making the experience as similar as possible among users. However, small variations are expected.

Collected Data: For each trial, we recorded the user's answer; for **C1** and **C2**, *Bump* or *Hole* while for **C3** was *yes* or *no*, depending on the perception of the stimulus. After each condition a questionnaire was introduced, for **C1** and **C2** we asked the following questions.

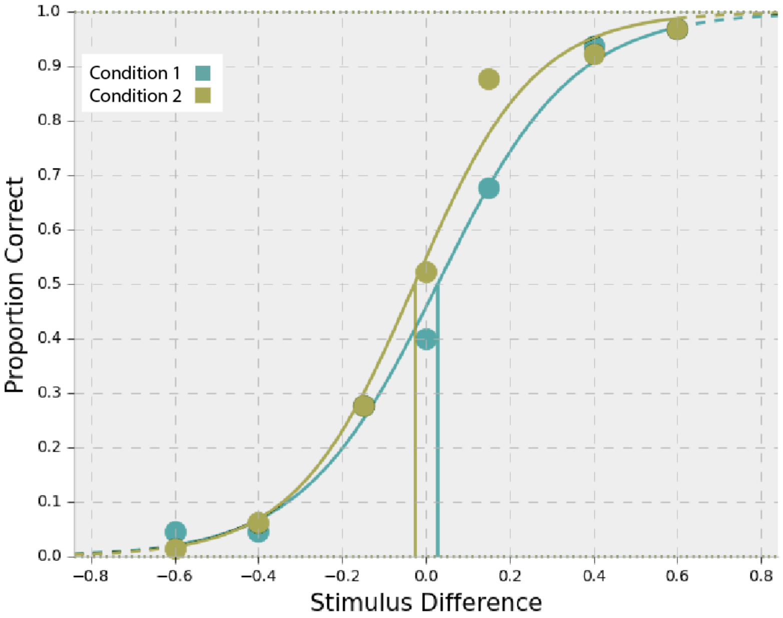
- **Q1.** When swiping, I felt the bumps and holes
- **Q2.** When swiping, it was easy to discriminate the shapes
- **Q3.** It was easier to perceive... **R.** 1.*Bump* - 7.*Hole*
- **Q4.** The wearable device in my finger distracted me from the primary task.
- **Q5.** Do you have any additional comment?

Condition **C3** had the following questions, also in 7-likert scale:

- **Q1.** When swiping, I felt the sticky and slippery surfaces
- **Q2.** When swiping, it was easy to discriminate the surfaces (sticky or slippery)
- **Q3.** It was easier to perceive... **R.** 1.*Sticky surface* - 7.*Slippery surface*
- **Q4.** The wearable device in my finger distracted me from the primary task.
- **Q5.** Do you have any additional comment?

At the end of the experience, some user accepted to share informally their thoughts about the sensations experienced.

Figure 3.20: Recognition rate, in yellow **C2**, in blue **C1**: the Point of subjective equality of the releasing condition is offset regarding the constant pressure condition.



3.3.4 Resuls and Discussion

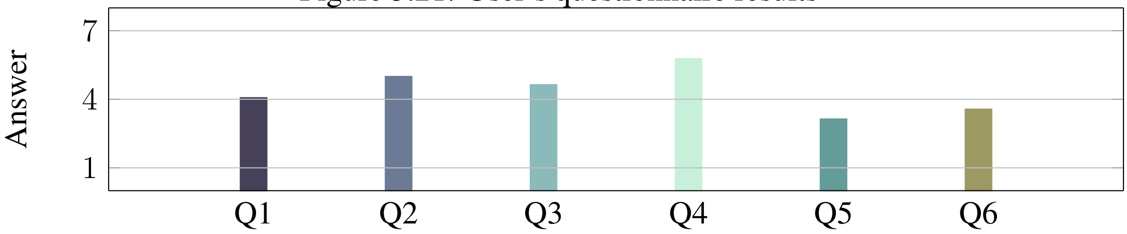
Experiment 1. Stiffness

We used a logistic regression model on the collected data to model the recognition rate of the stiffest piston for two independent variables **C1** and **C2** defined in the experimental design. The participants are considered a random effect in the model. We performed an One-Way ANNOVA analysis to validate our hypotesis: We found a significant difference between conditions ($p = 0.0000097$), which let us reject the null hypotesis (for $alpha = 0.05$). PSE for **C2** was -5.8% while for **C1** 3.6%. In the same way, Just noticeable difference was 19.8% and 20.1% for **C1** and **C2** respectively. The magnitude of these Just noticeable values difference is coherent with previous studies (TINGUY et al., 2018), (TIEST; KAPPERS, 2009).

Figure 3.20 shows the effect of **C2**: a visible offset to the left (less stiffness) is present in **C2**. This reflects the influence of the tactile stimuli on the softness discrimination; the presence of a preload when not touching and then releasing as long as the user presses reduce the perceived stiffness, in other words, the difference of pressure and not the actual pressure directly impacts on the subject's perception. Also, the difference of almost 10% between Points of subjective equality gives cues about the capability of this rendering mode.

Additionally, in table 3.2. The thresholds of perceived force versus recognition

Figure 3.21: User's questionnaire results



rate also reflects a shift in the perceived stiffness; There is a constant offset of $M = 8.93\% (SD = 0.05\%)$ between both curves (8.8% at 25%, 9.5% at 50% and 8.3% at 75%) while the Sensitivity remains very close ($\Delta Sensitivity = 4\%$). This shows that tactile stimuli impact mostly on softness/stiffness perceived but not in the accuracy or overall performance of the subjects. In this way, the use of the device rendering softness does not decrease the capacity of the users to discriminate between stiffness consistently.

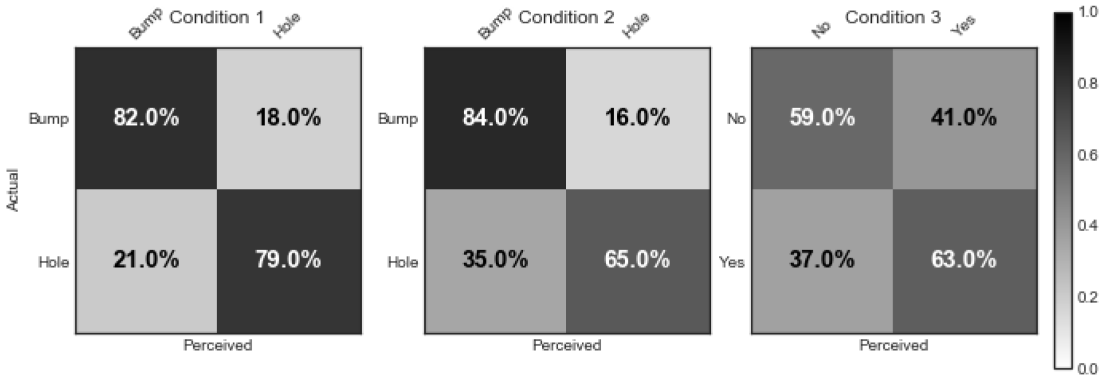
Subjective data: Figure 3.23 summarizes the answers to the questionnaire (7-point Likert scale). From the point of view of the users, the device barely contributed to the perception of stiffness (Q1, $M = 4.0$ $SD = 2.0$). Though, most of them agree that the combination of both feedback contributed to the perception of stiffness (Q2, $M = 5.0$ $SD = 1.7$). Also, from subjects, the tactile device barely affected their perception of stiffness (Q3, $M = 4.6$ $SD = 1.9$). In contrast, almost all of them agree that practicing improves the association of both sensations (Q4, $M = 5.7$ $SD = 0.9$). The device did not impact on their fatigue since they reported mostly not feeling tired (Q5, $M = 3.1$ $SD = 1.6$). Finally, subjects answered they were barely distracted by the actuation of our device (Q6, $M = 3.5$ $SD = 2.1$)

The results presented in this section suggest that our hypothesis about *softness rendering* is correct, nonetheless, an specialized study on this matter should help to complement the information presented in this work

Table 3.2: Recognition Performance

Condition	$Force_{perceived}$ at 25%	$Force_{perceived}$ at 50%	$Force_{perceived}$ at 75%	Sensitivity
C1	-16.4%	3.6%	23.2%	39.7%
C2	-25.3%	-5.8%	14.8%	40.1%

Figure 3.22: Confusion matrixes for the three conditions;



Experiment 2. Friction

For this experiment, we stored the answers of the users and contrasted them with the rendered stimuli. Figure 3.22 show the results of the three conditions; as visible at a glance, the first condition outperforms condition two when rendering holes. However, for the case of the bumps, there is a minimum difference. On the other hand, Condition 3 is not directly comparable with **C1** and **C2** but gives cues about the influence of the expectations on the perceived stimuli since it is the same rendering mode of **C2** but remarkably performed differently with a different question.

Table 3.3 gives relevant information about the performance obtained by the subjects in each condition. The best overall accuracy was obtained with **C1**, followed by **C2** and finally C3. Also in Precision and Recall, **C1** performed better. Results for **C3** are almost random regarding to *No* answers (Figure 3.22 Right) but also precision confirms this behavior; the amount of time that subjects labeled the surface 2 slippier than surface 1 was incorrect around 41.1% of the times ($Precision = 58.9$). In contrast, when asked about bumps and holes with the same rendering mode (C2), they got better results ($Precision = 76.5$).

A revealing metric is the Matthews correlation coefficient (MCC) which is related to the chi-square statistic for a confusion matrix: the values range from -1 to 1, being -1 a behavior opposed to the expected, 0 a random behavior and 1 a perfect classification. Condition 3 got an MCC score of 0.228, which means a behavior almost random. Condition 1

Table 3.3: Statistical metrics for all the conditions

Condition	Accuracy (%)	Error Rate (%)	Precision (%)	Recall (%)	MCC
C1	80.8	19.2	84.4	79.2	0.623
C2	74.6	25.4	76.5	65.4	0.492
C3	61.2	38.8	58.9	63.1	0.228

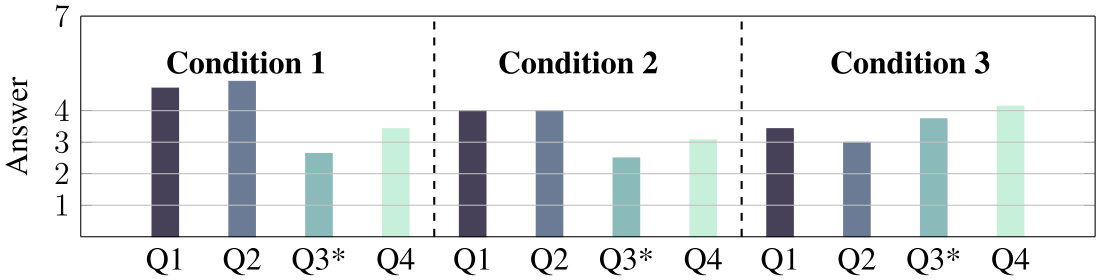
obtained the best results ($MCC = 0.623$), which indicates a positive correlation between rendered and perceived stimulus. Besides, **C2** shows a positive rendered-predicted stimulus but in a smaller amount, so, for the case of bumps and holes, the stimulation of slow adapting receptors instead of fast adapting receptors performed better. Contrasting **C2** and **C3** reveal a drop in the classification when there is a mismatch between expectation and stimulation.

From the subjective data we have for **C1** and **C2** that: Users barely felt bumps and holes, but they were more confident in **C1** (Q1 $M = 4.71$ $SD = 1.54$) than **C2** (Q1 $M = 4.00$ $SD = 1.96$). Also **C1** performed better when discriminating shapes (being aware that the shape is different) (Q2 $M = 4.92$ $SD = 1.54$) than **C2** (Q2 $M = 4.00$ $SD = 1.92$). When it comes to an specific shapes, both rendering methods were evaluated very simmilar: Bumps were easier to perceive than holes **C1** (Q3 $M = 2.64$ $SD = 1.44$), **C2** (Q3 $M = 2.5$ $SD = 1.34$). Regarding how much the device impacted over the task in terms of distraction, users perceived it was a bit more disctractive in **C1** (Q4 $M = 3.42$ $SD = 1.60$) than **C2** (Q4 $M = 3.07$ $SD = 1.49$) but not sifnificantly distractive. For condition 3 the results shows that subjects were not totally convinced about the presence of an sticky or slippy surface on the screen (Q1 $M = 3.42$ $SD = 1.39$). Also from their point of view, it was not totally easy to discriminate between surfaces (being aware that the surface is different) (Q2 $M = 3.00$ $SD = 1.17$) and this is confirmed in the Precision and Recall values for C3. According to the last point, users also stated that there were not surface easier to discriminate (Q1 $M = 3.71$ $SD = 1.58$). Surprisingly, they found the device more distractive on this task (Q1 $M = 4.14$ $SD = 1.79$) although the actuation principle is the same than C2.

Some subjects reported that discrimination of bumps in **C1** was straightforward, but they could not correlate the rendering of the hole with the actual sensation of touching a hole. So they assumed that not bump corresponds to hole but again, not being convinced about the rendering of holes in **C2** subjects mentioned that they were able to feel both holes and bumps.

In this case, we stimulated a region close to the fingertip but not close enough to interfere with the exploration of surfaces. Some subjects reported that they were not able to relate slippery sensation with the rendering because they were expecting the stimuli to be rendered at the fingertip, in the interacting area of the finger.

Figure 3.23: User's questionnaire results for friction test. *Questions with different answer meanings (No related with agreenment but with selection of bump, hole or type of surface)



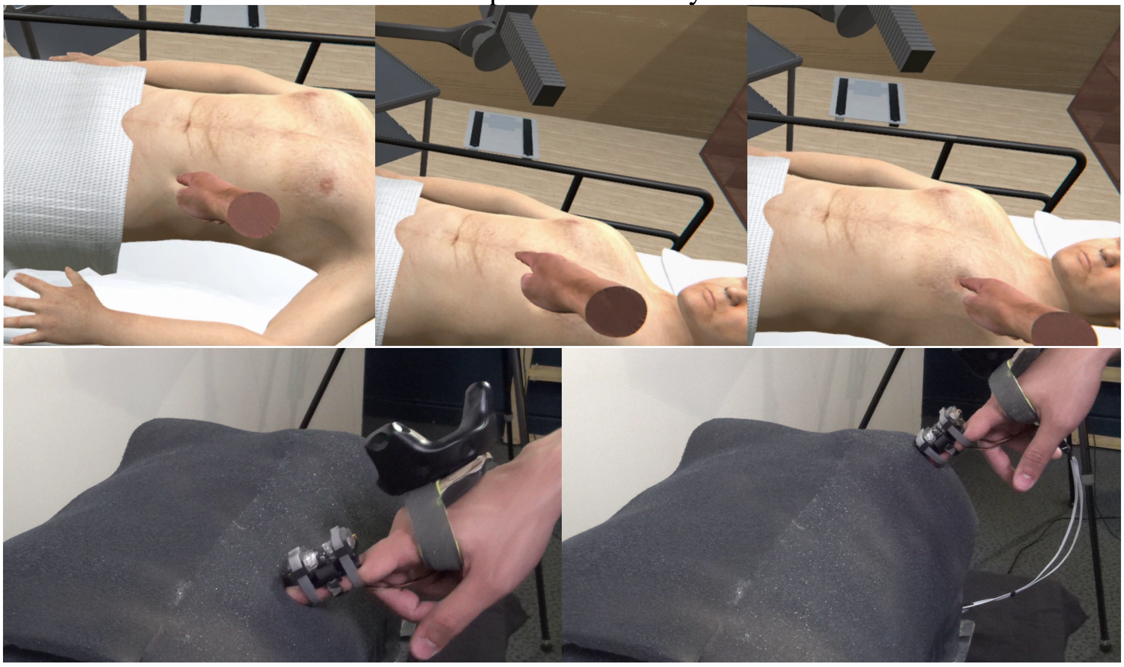
3.3.5 Use case

After the quantitative and subjective study of this approach to validate its perceptual consistency, we built a user case intended to show it working in context. We built a palpation scenario (Figure 3.13); we used an HTC VIVE for rendering the virtual scene, for tracking the position of the hand we used an HTC VIVE Tracker, and Unity 3D for building the immersive environment. The scene consists of a body laying on a surgery bed inside a medical room. In the real world, we placed a foam abdomen with constant stiffness. The user is asked to palpate the abdomen in several regions to check if the stiffness varies accordingly with the expected composition of the body (i.e., ribs are stiffer than the belly Figure (3.24)). The virtual body contains a deformable mesh which gives us the penetration deep (for this, we needed to synchronize the positions of virtual-tangible bodies)

Limitations

Our setup presented a small delay of about 1s which could impact negatively, mostly, in the friction experiment (where is most noticeable), but also in minor proportion in the stiffness test. The lack of self-tracking constrains the stand-alone use of the presented device. The incorporation of IMU sensors could help to have a rough track of the finger, but strategies as Multi-camera tracking becomes necessary to guarantee a reliable position tracking Finally, during the post-test conversation with a user, he said that the wearable device is hard to use for a long time, due to the preload. It is uncomfortable for long periods. In our opinion, it can be solved optimizing the design as well as changing the belt material for some softer fabric. Another factor impacting in this regard is the initial calibration, improving the calibration by finding an objective measure based on the

Figure 3.24: Palpation example using our methodology to render softness in a medical application: User is touching a soft part and then moving to a hard part: the device releases at different rates to render the different parts of the body.



finger size can lead to a better user experience

4 CONCLUSIONS AND FUTURE WORK

To the first chapter

Simulating: We presented a full Lagrangian method to simulate bidirectional phase shifting materials as a function of temperature. Solid, liquid and gas states were modeled with position based dynamics (PBD) as well as the phase-shifting processes of melting, solidification, vaporization, and condensation. Modifications to PBD density, viscosity and distance-constraints were introduced for the first time to simulate the thermal phenomena.

A latent heat model was also implemented to drive the transitions between the different states. A heat transfer model was adapted to comply with the law of thermodynamics that defines the heat transfer flow direction.

More complex phenomena such as sublimation, deposition or even plasma state were left out because they are less common. However, they are worth to be explored in future works. Another future development could explore the viscosity method proposed by Takahashi et al. (TAKAHASHI; NISHITA; FUJISHIRO, 2014) that could lead to soft melting/fusion transitions.

The core functions we proposed to simulate phase changes took, in the worst case, only 8% of the total timestep duration. This makes us believe that a parallelized version of PBD extended with our approach could equally be interactive with a considerably larger number of particles.

As our work is temperature-based, we need to obtain latent heat from temperature change. Introducing a fully energy-based model, although more computationally intensive, would be a more physics-based way to represent phase change and thermal phenomena in general. The proposed model, nevertheless, covers heat transfer by contact and, in our implementation, imitates convection. Future implementations must also include heat transfer by radiation, but none of these limitations impact the PBD constraints here proposed.

Better melting effects could be obtained using the very recent method by Weiler et al. (WEILER et al., 2018). However, their method carries more complex calculations.

Finally, the gas simulation could be improved developing new methods in SPH or PBF capable of handling smoke particles without the need for filling all the domain with atmospheric particles, since it is computationally impractical. In this topic, the work of

Ren et al. (REN et al., 2016) is worth to be explored.

Interacting: In this work we proposed and prototyped a particle-based system to sketch and simulate virtual objects with physical behavior in a VR environment. Results shown the performance achieved is sufficient to support 3D interaction and fluid animation of the objects. Two examples where physics is used are shown step by step in the figures 2.23 and 2.24.

The first informal experiments with target users shown promising results. The users, even if they hadn't any previous experience with VR, felt comfortable to move and interact in the virtual environment proposed. They also didn't report any cyberseekness symptom during the 40 minutes each one spent fully immersed.

Regarding the scenes sketched until now, they are interesting and sufficient to demonstrate the expressivity power of the sketching system proposed here. However, even if the main concepts about immersive sketching of dynamic objects have been verified, we are aware that the system need to be improved to be regularly used for artists.

As can be seen in the pictures along the interaction section, the rendering of the objects has a "legolized" appearance. An improvement of the render by smoothing the shapes would significantly improve the user experience. The marching cubes method generates smoother forms when the voxels are smaller. However, smaller voxels will demand more precision from the artists to sketch, as well as more computer power to simulate the objects behavior.

Path-constrained sketches are highly limited in terms of accuracy since the straight lines and geometries, in general, depends on the user ability to draw. Next works must move ahead introducing other tools, including the possibility of defining primitive bodies as cubes, spheres, rectangles, stars, etc. Additionaly, the possibility of creating bodies using extrusions and revolutions would expand the range of applications of our tool.

Another current limitation we have is the impossibility of taking a sketch and move it through the scene. The only post-sketching interaction between the user and the objects is the collision with the controllers. So the user is not able to accurately reposition a given object.

A fully immersive experience needs haptic and force feedback. This could improve the performance of digital artists when sculpting or modeling their artwork and is a possible future work.

Introducing gas particles is our next step to increase the capabilities of our model.

To the second chapter

Ultramotion:

We analyzed and developed a system to address the constrained workspace of Ultrasound phased-arrays by introducing new degrees of freedom by mean of a haptic robot.

We calculated, and fabricated an apparatus to drive the Ultrahaptics device in a virtual application. The apparatus was tested and evaluated by the haptics community on the EuroHaptics Conference 2018.

ULTRAMOTION enhances the Ultrahaptics capabilities by moving it according to the user needs. Thus, it can provide a more significant working area and new possibilities for applications and interaction.

This proof of concept device verifies that more degrees of freedom on the phased-array can open the window for new *less-constrained* applications. Not only rotational but also linear degrees of freedom should be explored for more dynamism of the system.

A guard system is fundamental to avoid collisions with the users when using bigger robots with more degrees of freedom, as recommendation this guard system should include procedural trajectory planning and collision handling.

Although this approach extends the workspace of ultrasound phased arrays, it constrains the velocity of movement of the hand: high velocities could lead to lost the tracking due to the mechanical limitations of movement. More advanced algorithms to control the robots must significantly improve the maximum velocity allowed.

We built this apparatus using an office device, the use of bigger robots as Kuka or Universal robots UR5 should expand even more the workspace while adding stability to the system.

Stiffness:

In this work, we presented a new stiffness rendering method using a wearable device. We rendered *softness* by applying a preload on the finger and then releasing as long as the users press the virtual object. Our approach cand dynamically alter the perceived softness of a given body in VR and natural reality.

Also, a method for rendering textures/friction was presented. We rendered bumps and holes using pressure (tightening and releasing the finger) and shear (Moving the belt from one side to another side of the finger). Also, we compared two different questions with the same rendering method to evaluate the impact on the experiment.

We conducted two experiments with users to validate our proposal and evaluated five different conditions (two in the first experiment and three in the second).

We considered a demo environment showing the strengths of our methodology, namely we designed a palpation scenario where the user touches an abdomen and chest to feel the internal composition of the body.

Although this study reveals that rendering of softness is possible by using our methodology, we think calibration and comfort of the design should be improved after any commercial distribution.

Also, the direct comparison with vibrotactile feedback in similar contexts worths to be explored.

This work then extends the range of applications of the device. We demonstrated the feasibility of modifying stiffness perception using the Hring. Additionally, for texture/friction rendering, we found a potential use which needs to be improved and further analyzed.

REFERENCES

AKINCI, N. et al. Versatile rigid-fluid coupling for incompressible SPH. **ACM Transactions on Graphics**, v. 31, n. 4, p. 1–8, 2012. ISSN 07300301.

Barrera Machuca, M. D. et al. Multiplanes: Assisted freehand VR drawing. **Adjunct Publication of the Annual ACM Symposium on User Interface Software and Technology (UIST'17 Adjunct)**, p. 1–3, 2017. ISSN 1463-9076.

BECKER, M.; IHMSEN, M.; TESCHNER, M. Corotated sph for deformable solids. In: CITESEER. **NPH**. [S.l.], 2009. p. 27–34.

BECKER, M.; TESCHNER, M. Weakly compressible sph for free surface flows. In: EUROGRAPHICS ASSOCIATION. **Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation**. [S.l.], 2007. p. 209–217.

BENDER, J.; MÜLLER, M.; MACKLIN, M. Position-based simulation methods in computer graphics. In: **EUROGRAPHICS 2015 Tutorials**. [S.l.]: Eurographics Association, 2015.

BERNDT, I.; TORCHELSEN, R.; MACIEL, A. Efficient surgical cutting with position-based dynamics. **IEEE computer graphics and applications**, IEEE, v. 38, n. 3, p. 24–31, 2017.

BISWAS, S.; VISELL, Y. Emerging material technologies for haptics. **Advanced Materials Technologies**, Wiley Online Library, p. 1900042, 2019.

BRISBEN, A.; HSIAO, S.; JOHNSON, K. Detection of vibration transmitted through an object grasped in the hand. **Journal of neurophysiology**, American Physiological Society Bethesda, MD, v. 81, n. 4, p. 1548–1558, 1999.

CARTER, T. et al. Ultrahaptics: multi-point mid-air haptic feedback for touch surfaces. In: ACM. **Proceedings of the 26th annual ACM symposium on User interface software and technology**. [S.l.], 2013. p. 505–514.

CHEN, C.-W. et al. Ontlus: 3d content collaborative creation via virtual reality. In: SPRINGER. **International Conference on Multimedia Modeling**. [S.l.], 2018. p. 386–389.

CLEARY, P. W.; MONAGHAN, J. J. Conduction modelling using smoothed particle hydrodynamics. **Journal of Computational Physics**, Elsevier, v. 148, n. 1, p. 227–264, 1999.

CULBERTSON, H.; KUCHENBECKER, K. J. Importance of matching physical friction, hardness, and texture in creating realistic haptic virtual surfaces. **IEEE Transactions on Haptics**, IEEE, v. 10, n. 1, p. 63–74, 2016.

De Tinguy, X. et al. Enhancing the stiffness perception of tangible objects in mixed reality using wearable haptics. **25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018 - Proceedings**, p. 81–90, 2018.

DIMENSION, F. Omega-7 overview. **Force Dimension http://www.forcedimension.com/omega7-overview**, 2013.

EROGLU, S. et al. Fluid sketching—immersive sketching based on fluid flow. In: IEEE. **2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)**. [S.l.], 2018. p. 475–482.

FAGIANI, R. et al. Tactile perception by friction induced vibrations. **Tribology International**, Elsevier, v. 44, n. 10, p. 1100–1110, 2011.

FARROKHPANAH, A.; BUSSMANN, M.; MOSTAGHIMI, J. New smoothed particle hydrodynamics (SPH) formulation for modeling heat conduction with solidification and melting. **Numerical Heat Transfer, Part B: Fundamentals**, v. 71, p. 299–312, 2017. ISSN 15210626.

FREEMAN, E. et al. Textured surfaces for ultrasound haptic displays. In: ACM PRESS. **Proceedings of the 19th ACM International Conference on Multimodal Interaction-ICMI 2017**. [S.l.], 2017. p. 491–492.

FRIER, W. et al. Using spatiotemporal modulation to draw tactile patterns in mid-air. In: SPRINGER. **International Conference on Human Haptic Sensing and Touch Enabled Computer Applications**. [S.l.], 2018. p. 270–281.

GAO, Y. et al. A novel fluid-solid coupling framework integrating flip and shape matching methods. In: ACM. **Proceedings of the Computer Graphics International Conference**. [S.l.], 2017. p. 11.

GAO, Y. et al. An efficient heat-based model for solid-liquid-gas phase transition and dynamic interaction. **Graphical Models**, Elsevier, v. 94, p. 14–24, 2017.

GIOIOSO, G. et al. Using wearable haptics for thermal discrimination in virtual reality scenarios. In: SPRINGER. **International AsiaHaptics conference**. [S.l.], 2018. p. 144–148.

GOLDSTEIN, E. B.; BROCKMOLE, J. **Sensation and perception**. [S.l.]: Cengage Learning, 2016.

GRAY, J.; MONAGHAN, J.; SWIFT, R. Sph elastic dynamics. **Computer methods in applied mechanics and engineering**, Elsevier, v. 190, n. 49-50, p. 6641–6662, 2001.

Hale, K. S.; Stanney, K. M. Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundations. **IEEE Computer Graphics and Applications**, v. 24, n. 2, p. 33–39, March 2004. ISSN 0272-1716.

HINCHET, R. et al. Dextres: Wearable haptic feedback for grasping in vr via a thin form-factor electrostatic brake. **The 31st Annual ACM Symposium on User Interface Software and Technology**, p. 901–912, 2018. Available from Internet: <https://doi.org/10.1145/3242587.3242657>.

HOCHSTETTER, H.; KOLB, A. Evaporation and condensation of sph-based fluids. In: ACM. **Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation**. [S.l.], 2017. p. 3.

HOSSEINI, M. et al. Exoten-glove : A force-feedback haptic glove based on twisted string actuation system. p. 320–327, 2018.

HOWARD, T. et al. Investigating the recognition of local shapes using mid-air ultrasound haptics. In: . [S.l.: s.n.], 2019.

HU, X.; ADAMS, N. A. An incompressible multi-phase sph method. **Journal of computational physics**, Elsevier, v. 227, n. 1, p. 264–278, 2007.

HUSSAIN, I. et al. A soft robotic supernumerary finger and a wearable cutaneous finger interface to compensate the missing grasping capabilities in chronic stroke patients. In: IEEE. **2017 IEEE World Haptics Conference (WHC)**. [S.l.], 2017. p. 183–188.

HUSSAIN, I. et al. A wearable haptic ring for the control of extra robotic fingers. In: SPRINGER. **International AsiaHaptics conference**. [S.l.], 2016. p. 323–325.

IHMSEN, M. et al. Implicit incompressible sph. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 20, n. 3, p. 426–435, 2014.

IHMSEN, M. et al. SPH Fluids in Computer Graphics. In: LEFEBVRE, S.; SPAGNUOLO, M. (Ed.). **Eurographics 2014 - State of the Art Reports**. [S.l.]: The Eurographics Association, 2014. ISSN 1017-4656.

JOHNSON, K. O. The roles and functions of cutaneous mechanoreceptors. **Current opinion in neurobiology**, Elsevier, v. 11, n. 4, p. 455–461, 2001.

JOHNSON, K. O.; YOSHIOKA, T.; VEGA-BERMUDEZ, F. Tactile functions of mechanoreceptive afferents innervating the hand. **Journal of Clinical Neurophysiology**, LWW, v. 17, n. 6, p. 539–558, 2000.

KEISER, R. et al. A unified lagrangian approach to solid-fluid animation. In: IEEE. **Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings**. [S.l.], 2005. p. 125–148.

KIM, Y. et al. CanvoX: High-resolution VR Painting in Large Volumetric Canvas. 2017.

KOVÁCS, P. T. et al. Tangible holographic 3d objects with virtual touch. In: ACM. **Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces**. [S.l.], 2015. p. 319–324.

LOOMIS, J. M. Tactile pattern perception. **Perception**, SAGE Publications Sage UK: London, England, v. 10, n. 1, p. 5–27, 1981.

LORENSEN, W. E.; CLINE, H. E. Marching cubes: A high resolution 3d surface construction algorithm. In: ACM. **ACM siggraph computer graphics**. [S.l.], 1987. v. 21, n. 4, p. 163–169.

MACKLIN, M.; MÜLLER, M. Position based fluids. **ACM Transactions on Graphics**, v. 32, n. 4, p. 1, 2013. ISSN 07300301.

MACKLIN, M.; MÜLLER, M.; CHENTANEZ, N. Xpbd: Position-based simulation of compliant constrained dynamics. In: ACM. **Proceedings of the 9th International Conference on Motion in Games**. [S.l.], 2016. p. 49–54.

MACKLIN, M. et al. Unified particle physics for real-time applications. **ACM Transactions on Graphics (TOG)**, ACM, v. 33, n. 4, p. 153, 2014.

MAEREG, A. T. et al. Wearable vibrotactile haptic device for stiffness discrimination during virtual interactions. **Frontiers in Robotics and AI**, Frontiers, v. 4, p. 42, 2017. ISSN 2296-9144.

MARTIN, S.; HILLIER, N. Characterisation of the novint falcon haptic device for application as a robot manipulator. In: CITESEER. **Australasian Conference on Robotics and Automation (ACRA)**. [S.l.], 2009. p. 291–292.

MCNAUGHT, A. D.; MCNAUGHT, A. D. **Compendium of chemical terminology**. [S.l.]: Blackwell Science Oxford, 1997.

MELI, L. et al. Combining wearable finger haptics and augmented reality: User evaluation using an external camera and the microsoft hololens. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 4, p. 4297–4304, 2018.

MELI, L. et al. Combining wearable finger haptics and augmented reality: User evaluation using an external camera and the microsoft hololens. **IEEE Robotics and Automation Letters**, IEEE, v. 3, n. 4, p. 4297–4304, 2018. ISSN 2377-3766. Available from Internet: <https://ieeexplore.ieee.org/document/8429065/>.

MICHAEL, H. **Virtual realism**. [S.l.]: Oxford University Press, 1998.

MINAMIZAWA, K. et al. Gravity grabber: wearable haptic display to present virtual mass sensation. In: ACM. **ACM SIGGRAPH 2007 emerging technologies**. [S.l.], 2007. p. 8.

MÜLLER, M.; CHARYPAR, D.; GROSS, M. Particle-based fluid simulation for interactive applications. In: EUROGRAPHICS ASSOCIATION. **Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation**. [S.l.], 2003. p. 154–159.

MÜLLER, M. et al. Position based dynamics. **Journal of Visual Communication and Image Representation**, Elsevier, v. 18, n. 2, p. 109–118, 2007.

MÜLLER, M. et al. Meshless deformations based on shape matching. **ACM transactions on graphics (TOG)**, ACM, v. 24, n. 3, p. 471–478, 2005.

MÜLLER, M. et al. Particle-based fluid-fluid interaction. In: ACM. **Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation**. [S.l.], 2005. p. 237–244.

NEALEN, A. et al. Physically based deformable models in computer graphics. **Computer Graphics Forum**, v. 25, n. 4, p. 809–836, 2006. ISSN 14678659.

NISHIMURA, N. et al. Wearable encounter-type haptic device with 2-dof motion and vibration for presentation of friction. In: IEEE. **2014 IEEE Haptics Symposium (HAPTICS)**. [S.l.], 2014. p. 303–306.

PACCHIEROTTI, C. et al. The hring: A wearable haptic device to avoid occlusions in hand tracking. **IEEE Haptics Symposium, HAPTICS**, IEEE, v. 2016-April, p. 134–139, 2016. ISSN 23247355.

PAN, J. et al. Real-time haptic manipulation and cutting of hybrid soft tissue models by extended position-based dynamics. **Computer Animation and Virtual Worlds**, Wiley Online Library, v. 26, n. 3-4, p. 321–335, 2015.

PROVANCHER, W. R.; SYLVESTER, N. D. Fingerpad skin stretch increases the perception of virtual friction. **IEEE Transactions on Haptics**, IEEE, v. 2, n. 4, p. 212–223, 2009.

QUILL • VR illustration and animation tool built to empower artists and creators. 2018. Available from Internet: <https://quill.fb.com/>.

REN, B. et al. Fast sph simulation for gaseous fluids. **The Visual Computer**, Springer, v. 32, n. 4, p. 523–534, 2016.

RUSSELL, S. Open dynamics engine v0. 5 user guide. **Computer Graphics**, v. 176, n. 2, p. 121–136, 2007.

SALAZAR, S. V. et al. Heat-based bidirectional phase shifting simulation using position-based dynamics. **Computers & Graphics**, Elsevier, v. 76, p. 107–116, 2018.

SAND, A. et al. Head-mounted display with mid-air tactile feedback. In: ACM. **Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology**. [S.l.], 2015. p. 51–58.

SCHECHTER, H.; BRIDSON, R. Ghost SPH for animating water. **ACM Transactions on Graphics**, v. 31, n. 4, p. 1–8, 2012. ISSN 07300301.

SCHMEDER, A.; FREED, A.; WESSEL, D. Best practices for open sound control. In: **Linux Audio Conference**. [S.l.: s.n.], 2010. v. 10.

SEO, J. H.; BRUNER, M.; AYRES, N. Aura garden: collective and collaborative aesthetics of light sculpting in virtual reality. In: ACM. **Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems**. [S.l.], 2018. p. Art12.

SHAO, S.; LO, E. Y. Incompressible sph method for simulating newtonian and non-newtonian flows with a free surface. **Advances in water resources**, Elsevier, v. 26, n. 7, p. 787–800, 2003.

SHERMAN, W. R.; CRAIG, A. B. **Understanding virtual reality: Interface, application, and design**. [S.l.]: Morgan Kaufmann, 2018.

SLATER, M.; USOH, M.; STEED, A. Taking steps: the influence of a walking technique on presence in virtual reality. **ACM Transactions on Computer-Human Interaction (TOCHI)**, ACM, v. 2, n. 3, p. 201–219, 1995.

SODHI, R. et al. Aireal: interactive tactile experiences in free air. **ACM Transactions on Graphics (TOG)**, ACM, v. 32, n. 4, p. 134, 2013.

SOLENTHALER, B.; PAJAROLA, R. Predictive-corrective incompressible sph. In: ACM. **ACM transactions on graphics (TOG)**. [S.l.], 2009. v. 28, p. 40.

SORGINI, F. et al. Neuromorphic vibrotactile stimulation of fingertips for encoding object stiffness in telepresence sensory substitution and augmentation applications. **Sensors (Switzerland)**, v. 18, n. 1, 2018. ISSN 14248220.

STEUER, J. Defining virtual reality: Dimensions determining telepresence. **Journal of communication**, Wiley Online Library, v. 42, n. 4, p. 73–93, 1992.

STOMAKHIN, A. et al. Augmented mpm for phase-change and varied materials. **ACM Transactions on Graphics (TOG)**, ACM, v. 33, n. 4, p. 138, 2014.

TAKAHASHI, T.; NISHITA, T.; FUJISHIRO, I. Fast simulation of viscous fluids with elasticity and thermal conductivity using position-based dynamics. **Computers and Graphics (Pergamon)**, Elsevier, v. 43, n. 1, p. 21–30, 2014. ISSN 00978493.

TIEST, W. M. B.; KAPPERS, A. M. Cues for haptic perception of compliance. **IEEE Transactions on Haptics**, IEEE, v. 2, n. 4, p. 189–199, 2009.

TILT Brush by Google. Available from Internet: <https://www.tiltbrush.com/>.

TINGUY, X. D. et al. Enhancing the stiffness perception of tangible objects in mixed reality using wearable haptics. In: IEEE. **2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)**. [S.l.], 2018. p. 81–90.

TOUCH, X. G. **Haptic Device**. [S.l.]: Accessed July, 2016.

WEILER, M. et al. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. **Computer Graphics Forum**, v. 37, n. 2, 2018.

WIERTLEWSKI, M. et al. A high-fidelity surface-haptic device for texture rendering on bare finger. In: SPRINGER. **International Conference on Human Haptic Sensing and Touch Enabled Computer Applications**. [S.l.], 2014. p. 241–248.

YEM, V.; KAJIMOTO, H. A fingertip glove with motor rotational acceleration enables stiffness perception when grasping a virtual object. p. 463–473, 2018.

YOH, M.-S. The reality of virtual reality. In: IEEE. **Proceedings Seventh International Conference on Virtual Systems and Multimedia**. [S.l.], 2001. p. 666–674.

YU, J.; TURK, G. Reconstructing surfaces of particle-based fluids using anisotropic kernels. **ACM Transactions on Graphics (TOG)**, ACM, v. 32, n. 1, p. 5, 2013.

ZUBRYCKI, I.; GRANOSIK, G. Novel haptic device using jamming principle for providing kinaesthetic feedback in glove-based control interface. **Journal of Intelligent and Robotic Systems: Theory and Applications**, Journal of Intelligent & Robotic Systems, v. 85, n. 3-4, p. 413–429, 2016. ISSN 15730409.

5 LIST OF PUBLICATIONS

Several articles with distinct contributions on Physically-based Animation, Human-Computer Interaction, Virtual Reality and Haptics were submitted throughout our research. The following is a complete list of our publications up to this time:

Journal Papers:

- ***Heat-Based Bidirectional Phase Shifting Simulation using Position-Based Dynamics***.
  VILLA SALAZAR, D. S., TICONA, A., NEDEL, L., MACIEL, A.
  2018, Computers & Graphics.
- ***Augmenting Softness and Roughness perception using wearable haptic***.
  VILLA SALAZAR, D. S., DE TINGUY, X., MARCHAL, M., PACCHIEROTTI, C., MACIEL, A.
  2020, For Submission.

Conference Papers:

- ***Phys-Sketch: Sketching 3D Dynamic Objects in Immersive Virtual Reality***.
  TICONA, A., **VILLA SALAZAR, D. S.**, NEDEL, L., MACIEL, A.
  2019, Computer Graphics International 2019.
- ***Towards Moving Virtual Arms Using Brain-Computer Interface***.
  SALAS, J., **VILLA SALAZAR, D. S.**, MACIEL, A., NEDEL, L., BARONE, D.
  2019, Computer Graphics International 2019.
- ***Evaluation of Visual, Auditory and Vibro-Tactile Alerts in Supervised Interfaces***.
  SOUZA, G., AMAYA, L., STEIN, V, **VILLA SALAZAR, D. S.**, TICONA, J., MACIEL, A., NEDEL, L.
  2018, 20th symposium on virtual and augmented reality.
- ***3DAthlon: 3D Gestural Interfaces to Support a 3-Stage Contest in VR***.
  GRANDI, JJ, DEBARBA, H, FRANZ, J, OLIVEIRA, V, TICONA, A, SOUZA, G, BERTI, I, **VILLA SALAZAR, D. S.**, NEDEL, L, MACIEL, A.
  2018, IEEE Conference on Virtual Reality and 3D User Interfaces.

Talks:

- ***Immersive environments: How to make the virtual, real?***. **VILLA SALAZAR, D. S.**
  2018, 1er Workshop creciendo en ciencia, Medellin, Colombia.
- ***Touching is believing: Experiences on blending physically-based animation and haptics*** **VILLA SALAZAR, D. S.**
  2019, Human-Centered Ubiquitous Media lab, Ludwig-Maximilian University, Munich, Germany.

Demonstrations:

- ***Balance Haptics Framework and Applications***. SOUSA CALEPSO, A., DE SOUSA S., G., MACHADO F., G **VILLA SALAZAR, D. S.**, MACIEL, A.
  2019, World Haptics Conference 2019, Tokyo, Japan.
- ***ULTRAMOTION: Enhanced workspace for ultrasound based haptics***. **VILLA SALAZAR, D. S.**, OLIVEIRA, V. A. de J., NEDEL, L., MACIEL, A.
  2018, EUROHAPTICS Conference, Pisa, Itay.
- ***3DAthlon: 3D Gestural Interfaces to Support a 3-Stage Contest in VR***. GRANDI, J., DEBARBA, H., OLIVEIRA, V., TICONA, A., FRANZ, J., SOUZA, G., BERTI, I., **VILLA SALAZAR, D. S.**, NEDEL, L., MACIEL, A.
  2018, 3DUI2018. Reutlingen, Germany.

Press:

- ***Estudantes da UFRGS vencem desafio de inovação no Japão***. Balance Haptics Framework and Applications,
  2019, RDCTV, Brazil.
  2019, UFRGS press, Brazil.
- ***Connessi attraverso il tatto: a Pisa il convegno internazionale***. ULTRAMOTION,
  2018, La Repubblica, Italy.
- ***Le nuove frontiere della robotica percettiva protagoniste alla “Eurohaptics 2018”***. ULTRAMOTION,

2018, La Nazione, Italy.

- ***Robotica percettiva, così si va oltre la realtà virtuale***. ULTRAMOTION, 2018, La Nazione, Italy.

APPENDIX A: TUTORIAL SHEET

This experiment is divided in 3 sections.

During each section, you must swipe and answer the question according to the tutorial shown at the beginning of the section.

You must swipe from the fingerprint signal to the opposite border of the screen

The swiping direction could change so, be aware of the arrows and the fingerprint signal

Take around 1s swiping

A pressure indicator is placed on the left side of the screen: try to keep it on the green region by applying a small force when swiping

Swipe once per surface unless the tutorial indicates something different.

The trial finishes one time you touch the corner of the screen so do not forget to swipe until you meet the border of the screen.

Answer the questionnaires after each stage; do not start a new stage without answering the previous section questionnaire.

H·REALITY

Inria
inventeurs du monde numérique

cnrs
dépasser les frontières

APPENDIX B: CONSENT FORM

H·REALITY

inventeurs du monde numérique

cnrs dépasser les frontières

Experiment Consent Form

Please tick the appropriate boxes

	Yes	No
Taking part in the study		
I have been informed of what I have to do in the study and have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	○	○
Safety		
I understand that this experiment involves the usage of a force feedback robot and a tactile wearable device, which, under the normal experiment conditions do not represent any risk.	○	○
Use of the information in the study		
I understand that the information I provide below will be used for quantitative purposes only and that no personal information such my name will be shared beyond the study team.	○	○

Signatures

By my signature below, I hereby certify that I have fully read this consent, understand its contents and have sufficient information to give this informed consent.

Name of participant [printed]   Signature   Date

______________________   ______________________   _____________

Researcher only

I have accurately informed the potential participant of the nature of the experiment and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

_________________________   ___________________   ________

Researcher name [printed]   Signature   Date

Participant No:________

APPENDIX C: QUESTIONNAIRE STIFFNESS EXPERIMENT

Demographic information

* Required

1. **Email address** *

   ____

2. **What is your age?** *

   Mark only one oval.

   ○ Under 18 years old
   ○ 18 to 24 years old
   ○ 25 to 34 years old
   ○ 35 to 44 years old
   ○ 45 to 54 years old
   ○ Over 54 years old

3. **What is yout gender?**

   Mark only one oval.

   ○ Female
   ○ Male
   ○ Prefer not to say
   ○ Other: ______________________________

4. **Highest degree or level of school the subject has completed. If currently enrolled, highest degree received**

   Mark only one oval.

   ○ No schooling completed
   ○ High school graduate
   ○ Trade/Technical/Vocational Training
   ○ Bachelor degree
   ○ Master degree
   ○ Professional Degree
   ○ Doctorate degree

5. **Dominant Hand**

   Mark only one oval.

   ○ Left
   ○ Right
   ○ Ambidextrous

6. **Do you have any previous experience using haptic wearable devices?**

   Mark only one oval.

   ○ Yes
   ○ No
   ○ Maybe

7. **Do you have any previous experience using haptic force feedback devices?**

   Mark only one oval.

   ○ Yes
   ○ No
   ○ Maybe

8. **Do you have any medical condition which could reduce your performance in the test?e.g. Loss of sensibility, Scar, Hyperhidrosis, Myopia, Hyperopia...**

Mark only one oval.

○ No, I don't have any problem or disability

○ Other: ____________________

Scope specific questions

9. **The haptic device on your finger contributed to the perception of stiffness.**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

10. **The combination of both cutaneous and kinesthetic sensations contributed to the perception of stiffness.**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

11. **The action of the tactile device affects my perception of stiffness**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

12. **Practicing improves the association of both cutaneous and kinesthetic sensations**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

13. **After the experiment, I felt tired.**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

14. **The wearable device in my finger distracted me from the primary task.**

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

15. **Do you have any additional comment?**

____

____

____

____

____

APPENDIX D: QUESTIONNAIRE FRICTION EXPERIMENT

Demographic information Friction

* Required

1. **What is your age? ***

2. **What is yout gender?**
Mark only one oval.

○ Female
○ Male
○ Prefer not to say
○ Other:

3. **Dominant Hand**
Mark only one oval.

○ Left
○ Right
○ Ambidextrous

4. **Do you have any previous experience using haptic wearable devices?**
Mark only one oval.

○ Yes
○ No
○ Maybe

5. **Do you have any medical condition which could reduce your performance in the test?e.g. Loss of sensibility, Scar, Hyperhidrosis, Myopia, Hyperopia...**
Mark only one oval.

○ No, I don't have any problem or disability
○ Other:

Skip to question 16.

PBH Condition.

6. **When swiping, I felt the bumps and holes**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

7. **When swiping, it was easy to discriminate the shapes**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly agree

8. **It was easier to perceive...**
Mark only one oval.

	1	2	3	4	5	6	7	
Bumps	○	○	○	○	○	○	○	Holes

9. **The wearable device in my finger distracted me from the primary task.**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

10. **Do you have any additional comment?**

Skip to question 11.

SBH Condition.

11. **When swiping, I felt the bumps and holes**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

12. **When swiping, it was easy to discriminate the shapes (Bump or Holes)**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	○	○	○	○	○	○	○	Strongly agree

13. **It was easier to perceive...**
Mark only one oval.

	1	2	3	4	5	6	7	
Bumps	○	○	○	○	○	○	○	Holes

14. **The wearable device in my finger distracted me from the primary task.**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

15. **Do you have any additional comment?**

Stop filling out this form.

SSS Condition.

16. **When swiping, I felt the sticky and slippery surfaces**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

17. **When swiping, it was easy to discriminate the surfaces (sticky or slippery)**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

18. **It was easier to perceive...**
Mark only one oval.

	1	2	3	4	5	6	7	
Sticky surface	○	○	○	○	○	○	○	Slippery surface

19. **The wearable device in my finger distracted me from the primary task.**
Mark only one oval.

	1	2	3	4	5	6	7	
Strongly disagree	○	○	○	○	○	○	○	Strongly agree

20. **Do you have any additional comment?**

Skip to question 6.

Untitled Section

Powered by
Google Forms