Contribution à l'étude du rendu et de la perception haptique d'objets virtuels déformables

Benoît Le Gouis

Thèse

UNIVERSITE
BRETAGNE
LOIRE

THESE INSA Rennes
sous le sceau de l'Université Bretagne Loire
pour obtenir le titre de
DOCTEUR DE L'INSA RENNES
Spécialité : Informatique

présentée par

Benoît Le Gouis

ECOLE DOCTORALE : *MathSTIC*
LABORATOIRE : *IRISA – UMR6074*

Contribution to the study of haptic perception and rendering of deformable virtual objects

Thèse soutenue le 21 Novembre 2017
devant le jury composé de :

Christian Duriez
Directeur de recherche, Inria Lille Nord Europe / Président
Sabine Coquillart
Directrice de recherche, Inria Grenoble Rhône-Alpes / Rapporteur
Fabrice Jaillet
Maître de conférences, Université de Lyon 1 / Rapporteur
Claude Andriot
Expert sénior CEA / Examinateur
Maud Marchal
Maître de conférences, INSA Rennes / Encadrante
Anatole Lécuyer
Directeur de recherche, Inria Rennes Bretagne Atlantique / Encadrant
Bruno Arnaldi
Professeur, INSA Rennes / Directeur de thèse

Acknowledgements

I would like first to thank my three advisors, Maud Marchal, Bruno Arnaldi and Anatole Lécuyer, for supervising me, for their numerous advices and for supporting me throughout this PhD. They gave me countless and precious advice on distinguishing between the essential and the superfluous, and telling me over and over again that the way to present my work is probably equally as important as the work itself in the context of scientific communication. This will most likely serve me well in the future.

I also would like to thank the members of the jury for taking the time to read the manuscript, for their numerous and insightful feedback on this work, and for their presence during my defense.

I would also like to thank Yoren Gaffary and François Lehericey for the two fruitful scientific collaborations. It has been really enjoyable to work and exchange together.

I also have to thanks the entire Hybrid team, as well as many members from the Mimetic for great moments spent together. Thanks in particular to Helen and David for many nice moments spent together, to Diane, Rens, Hakim, Carl, Romain Antoine and Emeric for nice Mercredi Gastronomie events, to Jussi for amazing Karaoke parties, and more generally all the others for really enjoyable moments.

Last, but not least, i would like to thank all my family for their support, in particular for the organisation of the amazing defense pot, which has relieved a great lot of pressure in the final weeks. And finally a great thanks to Caro for a never failing support and encouragements, and without whom nothing would have been possible.

Contents

Introduction 1

This PhD, entitled **Contribution to the study of haptic perception and rendering of deformable virtual objects**, presents research on improving haptic interaction with physically-based environments.

Since the release of consumer-grade haptic devices in the early 1990s, haptics has become a major tool to interact with virtual environments. From the greek *háptō*, "to touch", haptics relates to the sense of touch, and more specifically to the user interfaces that use the sense of touch. While first developed as a mean to display information through a different modality than the visual, with a display of surfaces [Minsky et al., 1990] or of volumetric force fields [Brooks Jr et al., 1990], haptic interfaces also allow great interaction with virtual environments by giving the possibility to act on it, and feel the effect that this action has generated. Haptics is a crucial part in many applications, such as training tools or teleoperations. For training tools, for example in surgery training, where a surgeon can train on a specific operation several times without requiring an individual to perform on, the haptic interaction must be as close as possible to the interaction in real conditions.

In order to achieve this fidelity to the interaction in the real conditions, objects must behave in the virtual environment as they would in reality. This can be performed using physically-based simulation. It computes the dynamics and behavior of all objects inside the environment, how they respond to external forces, to internal constraints, to contact, in real time. In particular, the simulation of deformable objects remains challenging, due to its inherent complexity caused by the need for the computation of the deformation. Simulating deformable objects is crucial, due to their omnipresence in most aspects of everyday life. For instance, simulating living tissue is an essential aspect of most training tools for surgery, character animation requires careful cloth simulation, and electric cable simulation is crucial for the virtual prototyping of large vehicles such as planes.

Most applications involving interaction, and in particular using haptic interaction, need to allocate resources as to maintain a certain quality (for example frame-rate) during the simulation, especially to maintain interactivity. This means that there is a trade-off to be made between performance and accuracy of the behavior of the simulation, with more accurate behavior being more costly and thus potentially leading to a loss of efficiency that might impair interaction. Adaptive methods have been proposed in order to offer more flexibility to handle this trade-off, by changing the allocated resources during the interaction according to what is really needed. The underlying premise of this PhD is that the perception of the user is a key factor in determining how the simulation should be simplified, and this especially if the behavior of the object is complex. Indeed since the interaction is performed by humans, the

perception that they have of the objects that they interact with is a major factor that must be taken into account in order to design a haptic interaction with virtual objects close to the interaction in real conditions. If a certain aspect of the simulation can be simplified without the user noticing it, there is an overall increase in efficiency without any loss in perceived accuracy, allowing to take the best of the trade-off. This enables the simulation of complex – thus interesting – behavior for the virtual objects, with the corresponding haptic interaction methods, with the best possible efficiency.

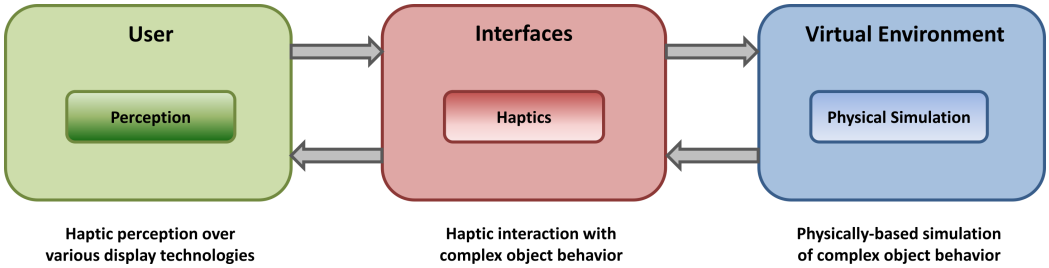


Figure 1.1 – The general haptic interaction loop: A user performs an action through an haptic interface, that has an influence on the virtual environment. Based on constitutive laws of the virtual environment, some feedback is provided back to the user using the interfaces. Challenges associated to each part of the loop for the interaction with complex objects are outlined.

Figure 1.1 depicts the haptic interaction loop that constitutes an interactive application, including the simulation of the virtual environment, with object interaction and simulation, the handling of interaction methods with the user, and the user perception. All three components must be efficiently handled in order to provide the best haptic interaction possible for VR (and to a lesser extent AR) applications. This PhD focuses on all three important aspects of this loop, *Physically-based Simulation*, *Haptic rendering*, and *Haptic perception*. Challenges associated to these aspects are hereafter described.

1 Challenges

In order to design a proper haptic interaction with complex objects, all aspects of the interaction loop have unique challenges to be addressed. First of all, the complex behavior of the objects must be simulated, which creates many simulation-related challenges. Once these objects are simulated, interaction requires proper corresponding haptic rendering methods, that creates additional constraints to the simulation challenges. Finally, since interaction usually involves a human participant, the human perception is a key factor that must be taken into account, especially with the emergence of various technologies for the display of virtual objects, that might influence perception.

1.1 Physically-based simulation of complex behavior from deformable objects

The design of an application for the interaction with objects of complex behavior starts with the physically-based simulation of such objects and behaviors. One of the first major aspects that must be taken into account is that – in VR – an application using physically-based simulation is by essence interactive, which means that the methods for the simulation have to be efficient enough to guarantee this interactivity. While essential even for the interaction with simple objects, this need for efficiency becomes crucial when the objects exhibit complex behavior, that often come at a price in terms of required computation.

The complexity of the behavior of an object can stem from many aspects. The haptic modality enables to feel internal properties of the object that might not always be visible from the surface. Such properties can include the heterogeneity of the object. Since most objects of our daily life are heterogeneous, being able to efficiently simulate them is crucial, and challenging. Another important factor of complexity in the behavior of the objects can come from the potential topology changes that might occur during the interaction with these objects. The most natural topology changes for deformable objects are associated to cutting and tearing phenomena. While cutting has received a lot of attention, in parts due to the need for it in surgery simulation, tearing simulation has been significantly less studied, especially coupled with haptics, and presents unique challenges.

1.2 Haptic interaction with complex behavior from the objects

Haptic devices have unique constraints that must be taken into account in order to design interactive applications. For instance, in order to ensure good haptic sensations, the refreshment rate should be around 1KHz for rigid interaction [Colgate et al., 1995], much more than the 30-200Hz frequency range required for visual feedback. This order of magnitude of difference means that methods aiming for haptic interaction must be specifically designed to be an order of magnitude more efficient, or find a workaround. In any case, the efficiency required for haptic interaction remains challenging, especially if the object exhibits complex – and computationally expensive – behavior.

Conversely, with potential applications in medical training, and surgeons performing simulated surgeries using haptic devices, there is also a great need for accurate feedback. In order for the training to be meaningful, the haptic behavior must be similar to real conditions, which comes in part from the simulation quality, but also from the haptic coupling method. When interacting with objects with complex behavior, this means that the simulation of this complex behavior has to come on par with appropriate rendering methods, which creates an additional challenge.

1.3 Haptic perception over different display technologies

Haptic perception in VR has received a lot of attention, with many results on stiffness perception, or on the visual predominance of vision over haptics. While not as common

as in VR, haptics is also used in AR applications, for example in medical training tools [Granados et al., 2017]. Most of these applications use the visual predominance over haptics to suggest physical properties on the object, from the sensation of "hardness" of the object with the superimposition of textures [Hirano et al., 2011] to the perception of weight of the object [Hashiguchi et al., 2014]. Perceptual studies suggest that some perception biases that were found in VR still stand in AR, with a potential difference in intensity. For example, the well-documented distance underestimation in VR also exists AR, but is slightly less important. The fact that this perception bias still exists in AR suggests a similarity in the perception of virtual objects in both environments. However, the difference in magnitude in both environments suggests an influence of the nature of the environment, virtual or augmented, over perception. Fully understanding haptic perception in AR remains thus a challenge.

2 Contributions

This manuscript addresses issues related to haptic interaction with physically-based environments. As described in Figure 1.2, the contributions associated to this PhD have a great emphasis on haptic interaction, in combination with the two important related domains, physically-based simulation and perception. The first **model**-oriented contribution addresses the issue of physically-based simulation of complex material, with the interaction with heterogeneous multi-resolution objects. The second contribution proposes a **simulation** method for the haptic tearing of deformable surface, with the full integration of complex haptic interaction with deformable objects undergoing topology changes. Finally, the last contribution studies haptic stiffness **perception** with two different display technologies, AR and VR.

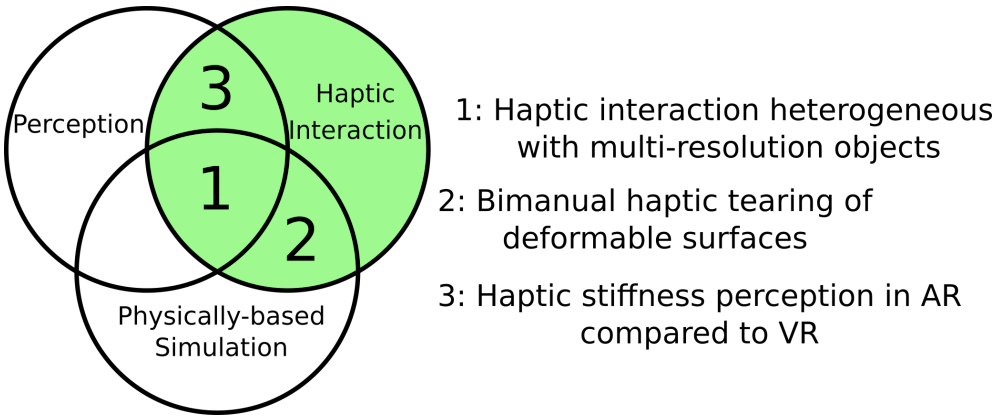


Figure 1.2 – Overview of the contributions. All 3 contributions have a great emphasis on haptic interaction, in combination with physically-based simulation, and perception

2.1 Haptic interaction with heterogeneous multi-resolution objects

Heterogeneous objects offer interesting challenges, both for simulation and for interaction. In order to simulate them efficiently, many methods coarsen the geometry

of the object, in a way that is visually close from the original object. However, the effect of this coarsening on the haptic perception during the interaction with these heterogeneous objects has received less attention. Our first contribution thus addresses the issue of geometric coarsening of heterogeneous objects, and more specifically the elasticity representation at the coarse resolution, and its effect on the haptic perception during the interaction with the coarsened object. We propose in the first hand a method redistributing the elasticity inside the object in order to facilitate the coarsening process. We then propose an evaluation of our method in terms of haptic perception during the interaction with heterogeneous coarse objects. This contribution proposes a model to address both the challenges associated to the physically-based simulation and the haptic rendering of complex heterogeneous objects. This method combines all three domains, haptic interaction, physically-based simulation of heterogeneous objects, and the evaluation uses a criterion based on perception knowledge.

2.2 Bimanual haptic tearing of deformable surfaces

Topology changes, and more specifically for thin deformable objects tearing, are a key aspect for the interaction with such objects. In our second contribution, we propose a complete simulation pipeline of interaction allowing haptic tearing of deformable surface objects. After the previous proof-of-concept model contribution, this simulation-oriented contribution targets complex haptic interaction with deformable objects undergoing topology changes. The simulation pipeline includes an efficient physically-based simulation of tearing for deformable surfaces, a collision detection method optimized for thin objects, as well as the haptic rendering method enabling bimanual tearing. This contribution is illustrated through several use-cases, including a complex scenario involving worn cloth.

2.3 Haptic stiffness perception in AR compared to VR

With complex objects simulated and proper haptic interaction, the last aspect to take into account is the user perception in order to build proper interaction methods. Understanding how the display technology for the environment (AR or VR) affects the haptic perception is a key factor to the design of interactive applications in AR. Our last contribution consists in a evaluation of haptic stiffness perception in AR compared to VR. It compares the perceived stiffness during the interaction with a virtual object superimposed on a real environment (AR context) and with the same object in a virtual environment (VR context). Collected data also allowed the evaluation of the used interaction strategy, the proper integration of the virtual object in AR, and the perceived similarity between the AR and the VR environments.

3 Outline

The manuscript is organized as follows :

Chapter 2 presents the related work on perception, physically-based simulation and haptics. This related work focuses on different biases that can affect perception, and on adaptive methods for the physically-based simulation and haptics, that allow to adapt the simulation accuracy depending on the needs.

Chapter 3 presents our contribution on multi-resolution heterogeneous deformable objects. It allows a geometric coarsening of the object that offers similar haptic perception.

Chapter 4 presents our contribution on haptic tearing of deformable surfaces. This contribution consists in a simulation and interaction pipeline with special attention to collision detection for surfaces, and tearing simulation.

Chapter 5 presents our contribution on the evaluation of stiffness perception in AR compared to VR. In a user study, we compared the haptic stiffness perception in AR and in VR, and showed that participants found objects in AR "softer" than in VR, and that they used a different interaction strategy based on the environment.

Chapter 6 concludes this thesis with a discussion as well as some perspectives.

4 Notations

In the following document, we use the following conventions:

- Constants are noted as normal letters: ρ, E
- Vectors are noted as bold lower-case letter: $\mathbf{x}$
- Euclidean norm for vectors is noted as such: $|\mathbf{x}|$
- Time derivatives from a time-dependent quantity $\mathbf{x}$ is noted as such: $\dot{\mathbf{x}}$
- Matrices are noted as bold upper-case letters: $\mathbf{M}$
- Dot-product between two vectors is noted as such: $\langle \mathbf{x} | \mathbf{y} \rangle$

Related Work

2

Contents

This chapter presents the related work to this PhD. As shown in Figure 2.1, the haptic interaction loop is composed of three major components, namely the *Physically-based simulation* of the virtual environment, the *Haptic Rendering* of this environment, and *Perception* that the user has of this environment. In order to build methods for the haptic interaction with a virtual environment, all three components need to be taken into account. This PhD aims at improving haptic interaction with deformable objects exhibiting complex behavior. One solution for this is to use adaptive methods, which is the reason why the related work both to physically-based simulation of virtual environments and haptics revolves around adaptive methods. An important requirement for these methods being to have minimal perceptual impact, the related work on perception mainly focuses on perception accuracy and perception biases. Section 1 discusses the related work associated with haptic perception, with a major focus on perception accuracy, and various perceptual biases. Then, section 2 presents the related work on physically-based simulation, with an initial presentation of different models for the objects, and then a discussion on adaptive methods. Finally, related work on haptics and corresponding adaptive methods is discussed in section 3.

1 Perception

1.1 Introduction

Interaction with any environment, real augmented or virtual, strongly depends on the perception that an user has of this environment. In order to build an application that

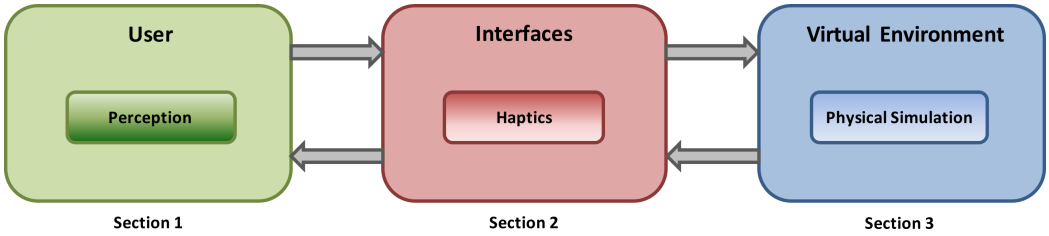


Figure 2.1 – The interaction loop displaying the major components for the haptic interaction with virtual environments, namely *Perception*, *Haptic Rendering* and *Physically-based simulation*

provides an optimal interaction, it is thus necessary to study the potential biases that can affect the perception of the user, as well as the limits of this perception. The following section first presents how haptic stimuli are perceived, from an anatomical point-of-view. It then presents unimodal perception, that is biases that affect one perception modality. Multimodal perception, which is the interaction between different modalities is then presented. Crossmodal perception, the influence of one modality over the perception of another modality, is presented last. Given the focus of this PhD on haptic interaction, the main studied modalities are haptic and visual.

1.2 The anatomy of haptic interaction

From a human point of view, haptic interaction relies on a sensorimotor loop, described in Figure 2.2. This loop is composed of 4 main components [Srinivasan and Basdogan, 1997]:

1. *The hand* of the user performs an action on an object, possibly a virtual object through an haptic interface, which creates **contact forces**;
2. *The sensors* from the hand and the limbs transform these contact forces into **tactile and kinesthetic information**. Tactile information is conveyed through the surface of the skin, whereas kinesthetic information comes from the position, movement and effort of the limbs. The receptors enabling both feedback information are hereafter described;
3. *The brain* analyses the received haptic information in order to determine the proper reaction, then issuing **motor commands** corresponding to this reaction;
4. *The muscles* convert this motor commands into **motion**, which makes the hand take a new position, acting on the virtual object, thus closing the sensorimotor loop.

1.2.1 Tactile perception

Tactile information is generated during the stimulation of three different receptors, located in the skin:

- *thermoreceptors* conveys information about the temperature of the objects. Two different receptors account for warmth and coldness perception;

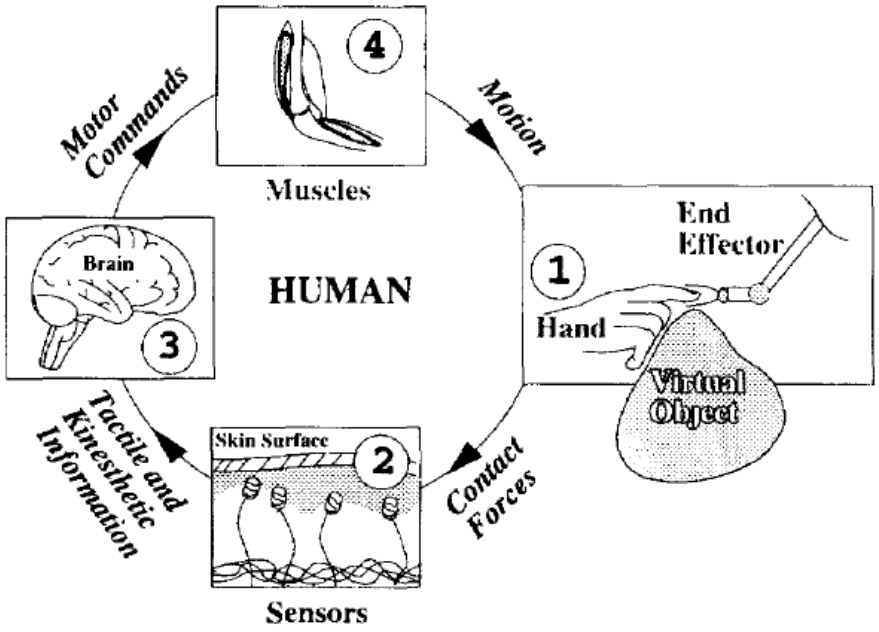


Figure 2.2 – Haptic interaction sensorimotor loop from the human perspective [Srinivasan and Basdogan, 1997]

- *nociceptors* are activated by mechanical, thermal or chemical stimuli, and indicate potential tissue damage by sending pain information;
- *mechanoreceptors* are activated by various types of mechanical stimuli, including pressure, vibration, skin stretch and texture change. The 4 different tactile mechanoreceptors are divided into two categories, 1) *slow-adapting receptors* with the Ruffini and the Merkel corpuscles, reacting respectively to skin stretch and sustained pressure, and 2) *fast-adapting receptors* with the Meissner and the Pacinian corpuscles, reacting respectively to skin stretch and sustained pressure.

Tactile perception of an object conveys information about spatiotemporal force variation, and can thus detect fine scale dimensions of an object, its texture, rubber-like compliance, and whether it is slipping relative to the skin [Srinivasan and Basdogan, 1997].

1.2.2 Kinesthetic perception

Kinesthetic perception, also referred to as proprioception, relates to the perception of position, orientation and movement of the limbs, as well as the associated forces [Srinivasan and Basdogan, 1997]. Proprioception is generated by two different types of receptors:

- *Tactile receptors*, with Pacinian and Ruffini corpuscles located in the tissue around the joints, provide information about joint angles and angular velocities;

- *Receptors in the muscles, tendons and joints* provide information about change in length (muscle spindles) or stretch (Golgi tendon organs) of the muscles.

Kinesthetic perception provides information about the large shape of the objects, spring-like compliances or the weight of the object. This PhD focuses on the perception of kinesthetic stiffness. Results on the accuracy of the human stiffness perception are hereafter described.

1.3 Unimodal perception

Most applications involving haptic interaction with a virtual environment provide some sort of visual information as well, but before studying the influence that the different sensory modalities can have towards each other, it is first important to study the senses by themselves, without the interaction from other senses. Many studies have showed, even when it is not the main focus of the study, that people sometimes make important perceptual mistakes, either because they are misled, or because they do not have the required sensitivity to properly feel the presented stimulus. For instance, [Greenish et al., 2002] asked users to recognize organic tissues just from haptic perception while cutting them, in order to compare the performance between real tissue and simulated tissue cutting and to evaluate the accuracy of their method. Results showed that users had bad results at recognizing even the real tissue with cutting, which indicates that most people did not have the proper sensitivity required for the task. This raises the question of the limitations of perception. To what extent are we able to perceive a given stimulus? This question can be addressed with two approaches, first from the biological limitations on perception, and second from the limitations of what the technology can convey, and whether it distorts the intended perception or not. We hereafter first present results from perceptual studies that focus on the discrimination capacity between two stimuli, with a perception threshold called the Just Noticeable Difference, and how this threshold can be used in applications in order to improve performances. Since most perception stimuli in VR are conveyed through an interface, it is also important to study how the interaction technology affects the perception, which is later on presented.

A Just Noticeable Difference

Senses are not infinitely accurate, and a difference between two stimuli might not be perceived if these two stimuli are close enough one from another. This corresponds to the Weber's Law, which states that, with ΔI the least noticeable difference in the intensity of a stimulus, and I the corresponding intensity, $\Delta I = k * I$, with k being called the Just Noticeable Difference (JND). There has been a lot of research to determine the JND for haptic perception. An initial study conducted by [Jones and Hunter, 1990] studied the JND in force and stiffness with a device attached to the wrist of users, the force being perceived at the arm level. The found JNDs were 10% in force, and 23% in stiffness. Tested stiffnesses were quite high, up to 6.26N.mm^{-1}. Other studies have focused on force and stiffness perception on other limbs, and other stiffness ranges. For instance, [Cholewiak et al., 2008] have studied the perception on the wrist, for stiffnesses between 0.2N.mm^{-1} and 3N.mm^{-1}. [Gurari et al., 2009]

studied the JND at finger level, for stiffnesses between 0.25N.mm^{-1} and 0.34N.mm^{-1}. Overall the JND range is found to be between 15% and 22% [Jones and Tan, 2013] for stiffness perception. This effect has several application, like compression for haptic data transmission for example. This has been used in [Hinterseer et al., 2006] and [Hinterseer et al., 2008] where new force value is transmitted only when the difference with the previous force value exceeds the JND threshold.

In a similar manner, some research has been conducted in order to study what the optimal geometrical resolution (see Section 2.3.1) for an object was so that the user would not feel the difference during the interaction in [Zhang et al., 2002], [Zhang et al., 2003] and [Payandeh et al., 2005]. A user study determined that, for the haptic interaction with an object, **there was no noticed difference between a precise model and coarser model**, meaning that there was no difference in the perceived force. The experimental protocol was to make the users interact with two objects in a row, and ask whether the users had felt any difference between the models. The visual model of the two objects was identical, in order not to include any visual bias. This results indicate clearly that, for the haptic sense, the precision of the sense is not absolute, and that there is room for optimization within normal range of simulation use.

B Technology-Induced Perception Biases

In addition to the aforementioned natural phenomena on perception, there exists biases related to the perception of virtual objects, in both augmented and virtual environments. For instance, distance perception is influenced by the displaying technology. Distances in virtual environments are underestimated [Rolland et al., 1995] [Knapp, 2001], with objects appearing closer than they really are. The estimated distances in VR can be 40% of the real corresponding distance. This phenomenon is explained in parts by the reduced number of context cues from which distance evaluation is performed. This phenomenon has also been observed in augmented reality, with a smaller underestimation [Jones et al., 2008] [Swan et al., 2006] [Swan et al., 2006]. Estimated distances in AR have been found to be around 80% of the real distance.

Due to its unique interaction between virtual and real objects, many factors can affect visual perception in AR. These elements are described in [Drascic and Milgram, 1996] and [Kruijff et al., 2010]. Five main elements are listed to be the cause of perception issues:

- **the real environment**, with aspects such as the structure of the environment, the colors of it, and the fact that it is indoors or outdoors;
- **the capture method** can have issues in the image resolution and filtering, the lens quality, the exposure, the color correctness and the capture frame-rate;
- **the augmentation method** can undergo problems related to registration errors, occlusion, layer interference and rendering and resolution mismatch;
- **the AR display device** with issues in the stereoscopy, the field of view, the display properties, the viewing angle, the color fidelity, reflections and latency;

- **the user** can have perception issues linked to individual differences, depth perception cues, accommodation, or disparity planes.

The most significant induced perception issues are visibility problems, scene distortion, meaning that the recognition of objects in the scene can prove difficult, and depth distortion, with a wrong evaluation of the depth of the augmented object inside the augmented environment.

For the haptic modality, technology can also induce perception biases. For instance, a delay in the haptic feedback has been shown to make the object feel softer, an phenomenon that has been experienced both in VR [Pressman et al., 2007] and in AR [Knorlein et al., 2009].

All these biases illustrate the fact that in order for an application to be has the expected perceptual behavior, the technological setup must be carefully taken care of, and proper implementation of the wanted behavior is not sufficient.

1.4 Multimodal Perception

While it is important to study senses by themselves, the interaction between them is also a major element for the perception. Senses can influence each other, which means that the perception of a virtual environment through different senses (multimodal perception) might be different than the perception of an environment through each sense taken separately. We first present multimodal enhancement, the fact that perceiving an environment with several modalities provides with better perception than the perception with separate modalities. We then present the interaction of haptic perception with first the visual modality, with a strong predominance of the vision over haptic perception, and interactions between the haptic and the auditory modalities.

C Multimodal Enhancement

For instance, having different senses involved for the perception of an environment usually provides better perception of the environment than separate perception with the different senses. This result has been shown for example by [Heller, 1982].

The studied feature for perception in this example was the perception of smoothness on a surface. Subjects were asked to classify objects by roughness based solely on vision, solely on haptic perception, and then using both senses. Given 3 objects of different smoothness, they were asked to say which was the smoothest. Results indicate that classification using vision or haptic separately gave similar results, but when combined, the results were significantly better. A possible explanation for this phenomenon given by the authors is that multimodal perception allows for optimal use of both senses during exploration.

In order to determine whether bimodal interaction would provide a richer perception, [Ballesteros et al., 2005] compared the perceptual space (hard-soft, dry-wet, etc) associated with a haptic exploration of objects with the perceptual space associated with haptic and visual interaction with the objects. The subjects were asked to classify objects based on their texture, without giving criteria for the classification, first solely based on haptic exploration, and then based on bimodal interaction. The goal of the study was to investigate whether the criteria for classification would be the

same under haptic exploration and bimodal interaction. The results indicate that with both interactions, subjects tend to use the same classification criteria, thus use the same perceptual space for object representation. This would seem to indicates that, even if multimodal interaction allows for a better accuracy, it does not necessarily introduce new perceptual dimensions.

Other research done by [Symmons and Richardson, 2009] has studied the case when haptic and vision had similar feedback. The study was focused on pattern recognition. Subjects were asked to recognize a letter by finger movement along the letter's shape. The provided visual information was a point from the approximate size of the moving finger tip. In this case, they observe no increase in performance when vision and haptic are matched, which indicates that when the information is identical, there is no clear advantage to a multimodal perception.

These results were obtained studying the interaction with real objects, with matching information coming from different senses. The use of simulation and haptic devices has also allowed to study the impact of sense from one to another, when the sensory information provided to the different senses do not match.

D Visual dominance over haptic

The influence of vision on haptic perception has been studied for various problems. One of the challenges for haptic interaction is to be able to estimate the stiffness of the interacting elements. [Srinivasan et al., 1996] studied the influence of vision for the estimation of the stiffness of a virtual spring. The subjects were asked to manipulate two virtual springs, and state which one was the stiffer spring. A visual bias was introduced so that the displacement produced by an applied force on a spring would not be directly linked to its physical stiffness. Instead, virtual springs were given two different values of stiffness each, a constant one for force feedback, and another one varying between the two physical stiffness values in order to compute the displacement for a given force. For extreme values, the spring elongation was inverted between the two springs for the same force value. Results indicate that the visual elongation of the spring strongly influenced the subjects for the classification, which would indicate that on this case, the visual sense would prevail on the haptic sense.

Similar results were obtained in [Kumazaki et al., 2007] for the problem of haptic length perception. The subjects were asked to estimate the distance traveled by their hand, with the vision of the hand provided through a video screen, with an edited image of the hand. When the hand is rendered normally, but with different distance traveled, the estimated distance tends to match the one from the visual input. However, tests were performed with the image of the hand heavily changed by noise. When the noise is important (more than 60%), haptic information of distance prevails during the estimation. This indicates that **when the subject can rely on vision, visual information will have more weight in perception**.

[Yamamoto and Okamura, 2007] used a drawing task to study the effect of haptic error on the on subjects. The task was to draw a shape, possibly with incorrect haptic feedback. The study shows that performances were not affected if the error was below a certain threshold (in this case, a 5 degrees orientation error). However, erroneous

haptic feedback provided better results than no haptic feedback.

In order to detect how the stiffness perception was influenced by vision, [Sulzer et al., 2007] conducted a study in which subjects were asked to move the hand from one position to another on the surface of a ball. The balls had different stiffness, and visual clues were also given on the ball stiffness (the indicated stiffness being possibly different from the actual stiffness). The main focus was on the adaptation of the behavior based on the given visual information. Results indicate that subjects adapt to visual information, but not uniformly to the difference between haptic and visual stiffness. For instance, there was more adaptation to low stiffness visual information than for high stiffness.

Several other previous works have focused on how the perceived haptic properties of a real object could be changed by a visual superimposition of information on this object. Hirano et al. [Hirano et al., 2011] have notably superimposed textures associated with different levels of hardness on a real object, successfully influencing the perception of hardness of this object. Similar methods have been proposed to influence a "softness" perception [Punpongsanon et al., 2015] or the perceived weight of an object [Hashiguchi et al., 2014]. In a purely AR context, Jeon and Choi [Jeon and Choi, 2008] have also shown how adding a force-feedback during interaction with a real object could modulate the stiffness perceived by the user.

E Haptic and sound interaction

First, [DiFranco et al., 1997] studied the effect of sound usually associated with a specific stiffness on the estimation of stiffness of surfaces. Subjects were asked to rank surfaces based on perceived stiffness. At first, as a baseline, all surface were from same stiffness, with sounds associated to various stiffness. As expected, subjects ranked the stiffness based solely on auditory clues. Other experiments were performed with random association between haptic stiffness and sound stiffness. Subjects that had not performed the first experiment also based their judgment on auditory clues rather instead of haptic clues. However, expert subjects were more prone to rank based on actual stiffness. This result indicates that sound can also influence haptic, based on usual association to known stiffness.

Other studies have investigated the influence of sound on roughness perception. [McGee et al., 2001] and [McGee et al., 2002] performed an experiment where subjects were asked to move a haptic device on a textured surface. Sounds were played when the haptic device was close enough from a peak in the texture. The test was to determine whether two presented surface had the same roughness or not. During the experiment, the frequency of the sound was changed, in order to indicate a potentially different roughness. Results indicate that the subjects were influenced by the sound.

These results indicate that, though not as clearly as vision, sound clues can be used to influence perception of haptic properties of an object.

1.5 Crossmodal perception

Crossmodal perception occurs when the perception of stimulus with a given sense enhances, or even completely produces the perception of the stimulus by other senses. The main difference with multimodal methods lies in the fact that the perceived stimulus is not directly linked to the other modality, but affects it nevertheless. For instance, [Vroomen and Gelder, 2000] have studied the interactions between vision and audition. The focus of the study was the recognition of a visual pattern made possible by the emission of a sound. Subjects were asked to recognize a visual pattern on a screen on which patterns changed very quickly. A sound was played for each displayed pattern. The recognition was much improved by a change in tone during the display of the target pattern. Further experiments were made to ensure that the better recognition was not a side effect of an expected auditory pattern, but rather on a crossmodal of an auditory on vision.

Similarly, [Frassinetti et al., 2002] studied the effect of a played sound on the perception of a visual signal. The subjects were placed in front of several light sources, with a loudspeaker for each source. Light impulse were regularly emitted, with two different possibilities. Either only red distractor lights were lit, or before the red distractors a green light was lit, the positive case to detect. A sound was also emitted, either at the same moment as the visual impulse, or about 500ms before, and coming from any of the speakers. The detection of the actual signal was significantly improved when the sound was emitted from the corresponding speaker, at the moment of the light impulse. Sound at other locations, or before the visual signal had no significant consequence. This phenomenon can be explained by the fact that the neurons for the detection of such impulses are also used for auditory perception as well.

Interaction between vision and haptic is not limited to multimodal perception. It is for instance possible to induce a haptic perception with visual information. [Spence et al., 2004] have studied the effect of visual-tactile crossmodal distractors. The subjects are given a cube of foam in each hand, with two vibro-tactile devices on each, intended for contact with the thumb and the index, and visual distractors near each vibro-tactile device. Subjects are asked to perform an action based on what device is vibrating (up for the index, down for the thumb). Distractor lights were shown to strongly influence the perception of the vibrating device.

A crossmodal approach using visual dominance over haptics called *pseudo-haptic feedback* was also proposed [Lécuyer, 2009] . The authors notably showed how playing with visual feedback enables to simulate a wide range of stiffness sensations when using a passive elastic device [Lécuyer et al., 2000]. The researchers noticed a perceptual offset between the perception of a real spring and the perception of such a pseudo-haptic spring simulated with visual feedback. The pseudo-haptic spring was globally underestimated compared to the real spring. Using a psychophysical method, they found that the perceptual offset (or Point of Subjective Equality) was on average equal to 9%. Interestingly enough, other perceptual biases have also been observed regarding haptic perception of stiffness such as a depth or perspective effect [Wu et al., 1999]. This effect implies that objects located at a farther distance are perceived as stiffer.

1.6 Conclusion

In this section, we have discussed about different aspects of haptic perception, first by itself and then in combination with other sensory modalities. First of all, there are limitations in the natural perception sensitivity, and two haptic stimuli need only be quite similar to be perceived as equal, which can lead to simplification in the simulation, and thus better performances. The JND is indeed a useful tool in order to evaluate the perceptual influence of a simplifying method for the simulation. Furthermore, interaction between senses can also be used for haptic applications. For some interactions, one sense can prove to be really predominant compared to the others involved. Stimuli involving one sense can even be used to create the sensation of a stimulus from another sense, with crossmodal applications. However, this sensory predominance can lead to unwanted – or just unexpected – shifts in perception, which must be thoroughly investigated. Perception is thus an important aspect to take into account for the design of strategies for simpler simulation with equivalent sensation to the user.

2 Physically-based Simulation

2.1 Introduction

Physically-based simulation computes the dynamics and the behavior of the objects inside a virtual environment, how they respond to external forces, to internal constraints, to contacts, in real time. Most simulation methods aim at achieving a physically accurate (or at least a visually plausible) behavior. In the context of Virtual and Augmented Reality, interaction is an important factor, meaning that the simulation pipeline involves interfaces, and must handle interaction. Interaction means that the methods must be efficient enough that an action of the user is handled with no noticeable delay, and thus required efficient methods.

The general simulation loop, depicted in Figure 2.3 is as follows: the user performs an action, through the sensori-motor interfaces. The first step to solve the effect of that action is to detect what objects in the environment interact together, and the main interaction between object occurs at contact. The first step is thus collision detection. With this contact between elements, the physics behavior of the objects in the scene can be computed. While different physical models exist, for rigid objects, deformable objects and fluids, each having specific challenges related to collision detection, collision response and overall dynamics, this PhD focuses on deformable objects. Once the physical behavior of all the objects in the scene has been computed, feedback based on the environment state must be computed, and provided to te user via the interface, closing the interaction loop.

In this section, we present first simulation for deformable object. For more information on rigid objects simulation, please refer to [Bender et al., 2014]. The main elements of fluid simulation can be found in [Bridson and Müller-Fischer, 2007] for the dynamics of the fluid, and in [Wojtan et al., 2011] for the surface tracking of the fluid.

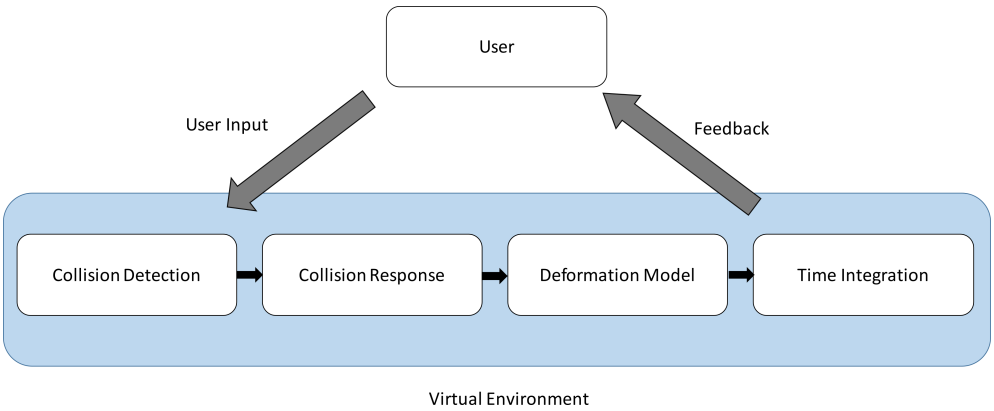


Figure 2.3 – The general simulation loop: the user performs an action through sensori-motor interfaces. Collision between elements is detected. Based on the contact information, the behavior of the elements in the scene is computed, including the collision response and the object deformation, usually obtained through a differential equation that must be integrated over time. Once the behavior of the environment computed, feedback can be sent to the user, closing the interaction loop. Adaptive methods can improve the performances of most aspects of the loop.

Simulation for deformable objects includes the computation of the deformation, with elements from the Continuum Mechanics, and from the two main representations, Finite Element Method and Mass-Spring Models. A brief presentation of collision detection, collision response and time integration methods is also presented, as important parts of the simulation loop. Finally, the last part of this Section describes adaptive methods, with a special focus on adaptive methods for heterogeneous objects, which offer additional challenges in terms of simulation, and perception.

2.2 Physically-based Simulation of Deformable Objects

2.2.1 Deformation Models

Deformable objects represent a challenge when it comes to simulation. Additionally to the challenges related to rigid object simulation, deformation of the object must be taken into account, which makes each step more computationally challenging, collision detection, collision response and time integration. Early work has been performed in this field in [Terzopoulos et al., 1987]. A complete survey on deformable objects simulation can be found in [Nealen et al., 2006].

In order to deform objects, a model for the deformation is needed. The representation of the object is a first step in this choice of deformation. There are traditionally two main categories of representation for a deformable object, discrete representation and continuous representation. In the discrete representation, matter is discretized into a finite set of punctual objects sharing the total mass of the object, and linked together by springs. Deformation can then be derived using the classical mechanics equations on springs. This representation is called the Mass-Spring representation.

In the continuous representation, the object is discretized into a finite set of elements, such as tetrahedra, that form a mesh of the objects. The deformation is

performed using Continuum Mechanics equations, discretized on the vertices of the mesh. It is important to note that the representation is continuous, since the the mesh is only a discretization for the equations, but the matter is still considered continuous inside the mesh elements. This method to solve a complex partial differential equations system (such as the Continuum Mechanics equations for the deformation of an object) from the solutions of the equations at a finite set of points (the vertices of a mesh) is called the Finite Elements Method (FEM).

We first present an introduction to Continuum Mechanics, then describe further the two representations.

A Continuum Mechanics

Continuum mechanics studies the behavior of matter, considered as continuous. It provides equations for physically accurate deformation of a body. In order to achieve this goal, it is important first to be able to know at each moment the displacement of all material point. The displacement $\mathbf{u}$ for a point p is formally the difference between the current position $\mathbf{x}(p)$ of the point and its rest position $\mathbf{x}_0(p)$

$$\mathbf{u}(p) = \mathbf{x}(p) - \mathbf{x}_0(p) \tag{2.1}$$

With this displacement field, we can characterize the deformation of the body, called the strain in mechanics. The strain measures the relative elongation of the body, for all dimensions. In 3D, the strain is related to the displacement field using a tensor of order 2:

$$\epsilon_G(\mathbf{x}, t) = \frac{1}{2}(\nabla\mathbf{u}(\mathbf{x}, t) + [\nabla\mathbf{u}(\mathbf{x}, t)]^T + [\nabla\mathbf{u}(\mathbf{x}, t)]^T\nabla\mathbf{u}(\mathbf{x}, t)) \tag{2.2}$$

$$\epsilon_C(\mathbf{x}, t) = \frac{1}{2}(\nabla\mathbf{u}(\mathbf{x}, t) + [\nabla\mathbf{u}(\mathbf{x}, t)]^T) \tag{2.3}$$

ϵ_G is called Green-Lagrange non-linear tensor, and ϵ_C is its linearized version.

When a body is deformed, internal forces are created, aiming at bringing the body back to its initial state. These forces are the internal stresses of the body. They depend on the actual strain, and on the stiffness of the body. For a Hookean body, the relation between the stress σ and the strain is linear:

$$\sigma(\mathbf{x}, t) = E\epsilon_C(\mathbf{x}, t) \tag{2.4}$$

where E is the so-called Young's modulus, representing the stiffness of the body.

B Mass-Spring Models

The mass-spring representation has been widely used due to its computational efficiency. The object is represented as a set of material points with a mass, bound together by massless springs [Chen et al., 1998]. Different springs can be used to model different behavior, such as shear, flexion, or even plastic deformation.

The object's dynamic is completely described by the respective positions $\mathbf{x}_i$ and velocities $\dot{\mathbf{x}}_i$ of each mass point i. In order to know the effect of the forces $\mathbf{f}_i$ exerted on each mass, Newton's second law is used:

$$m_i\ddot{\mathbf{x}}_i = \mathbf{f}_i \tag{2.5}$$

This gives for the entire system the following equation:

$$\mathbf{M}\ddot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) \tag{2.6}$$

with $\mathbf{M}$ the diagonal mass matrix of the entire object. The resolution of this system provides the dynamics for the entire object. This model is quite simple, since it only consists in an ODE system to solve. The forces created by the spring are classical elastic spring forces:

$$\mathbf{f}_{ij} = k_{ij}(|\mathbf{x}_j - \mathbf{x}_i| - l_{ij})\frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{2.7}$$

with k_{ij} the stiffness of the spring, and l_{ij} its rest length. In order to induce relaxation due to energy dissipation during movement, damping can be introduced as following:

$$\mathbf{d}_{ij} = k_d\langle\mathbf{v}_j - \mathbf{v}_i|\frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}\rangle\frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|} \tag{2.8}$$

with k_d the damping coefficient.

This model, while very efficient, suffers several limitations. First of all, since it does not rely on physical properties of the object, it lacks accuracy. Furthermore, this representation does not take into account volume properties, leading to potentially important changes in volume during the simulation. The model is also non convergent, meaning that given a situation, it does not always converge to the right solution, and this can lead to important instabilities during the simulation.

C Finite Element Method

In order to have a more precise simulation, the Continuum Mechanics equations are used to compute the deformation of the object, and solved using the FEM [Chung, 1996]. The main idea of FEM is to discretize the equation on a finite representation of the object, but keeping the continuous state of matter. The object is discretized into a – often tetrahedral – finite mesh, but the matter is still considered continuous inside the elements. For an extensive presentation of all the FEM simulation process, please refer to [Bro-Nielsen, 1998].

For elastic materials, the equation of deformation is

$$\rho\ddot{\mathbf{x}} = \nabla.\sigma + \mathbf{f} \tag{2.9}$$

with ρ representing the density of the material, and $\mathbf{f}$ the external forces to the object. This continuous differential equation can be discretized at a finite set of locations that form the mesh of the object. Solving this system of equations can be seen as an optimization problem, using the Galerkin method [Hunter and Pullan, 2001]. Galerkin

approach solves an equation with infinite solution space by projection on a finite subspace, and giving exact solution on this projection. The approximation error is orthogonal to the considered subspace.

For more simplicity, other approaches rely on an explicit representation for FEM simulation. Instead of having a global equation, discretized on the mesh, the forces are directly considered at the mesh vertices, as well as the mass. With this representation, the equation for deformation can be explicitly given as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.10}$$

$\mathbf{M}$ is the mass matrix, associating to each element its mass, $\mathbf{D}$ is the damping matrix, and $\mathbf{K}$ is the stiffness matrix, enclosing material properties of the material such as elasticity. $\mathbf{Ku}$ represents the internal forces induced by the deformation. This linearized version of the FEM can lead to important artifacts when large deformation are applied.

In order to address this problem, the rotational part of the deformation field is extracted before computing the induced forces [Hauth and Strasser, 2003] [Müller and Gross, 2004] [Nesme et al., 2005]. Noting $\mathbf{R}$ this rotation, the value for internal forces in the tetrahedron is

$$\mathbf{f} = \mathbf{R}\mathbf{K}(\mathbf{R}^{-1}\mathbf{u}) \tag{2.11}$$

The rotation is obtained through polar decomposition of the transformation matrix between deformed and rest state. This method, called corotational FEM is used in most simulators as of today.

While this method provides good accuracy, it requires a lot of computation, making it more difficult to simulate complex objects at interactive rates. In order to improve the performances, [Courtecuisse et al., 2010] propose an implementation that allows for GPU use. The corotational computation of force being independent between different elements, it can thus be implemented in parallel. However, the forces are computed for vertices that can be shared between several elements. After force computation, the resulting force for each vertex is computed based on all the computed forces in the previous phase. This method allows for a significant speedup of the simulation framerate.

2.2.2 Collision Detection

Collision detection computes for each pair of objects whether they are in contact or not. A simple way to do it is to test intersection for each pair of elements of both meshes whether they are in contact or not. This requires important computational performances, since the number of pairs of elements between two important meshes is really important. In order to be able to perform efficient collision detection, the process has been separated in several phases, the broad phase, the mid phase and the narrow phase. During the first phase, the broad phase, the simulation space is separated in regions, and every object is assigned to the regions in which it lies, as explained for example in [Overmars, 1992]. If the regions of two objects do not overlap, there is no need for collision detection, since they cannot intersect. The second phase, the middle phase, reduces further the number of potential pairs to check.

A popular approach for this is the use of Bounding Volume Hierarchies (BVH). The object is decomposed into a set of recursive bounding volumes, and further intersection is then performed between these volumes. If two volumes intersect, the test is then performed with underlying volumes, and this recursively until the bottom level of the hierarchy is reached. Collision detection only needs to be performed on elements located in intersecting bottom level volumes. There are multiple hierarchy types, such as sphere trees ([Palmer and Grimsdale, 1995], [Weller and Zachmann, 2009]), *kd*-trees ([Klosowski et al., 1998]) or octrees ([Bandi and Thalmann, 1995]). The final step, the narrow phase, checks into bottom level volumes what elements to check for intersection. For extensive reviews on collision detection methods, please refer to [Teschner et al., 2005] and [Avril et al., 2009].

2.2.3 Collision Response

Once the collision is computed, the next step is to compute how the collision affects the object kinematics. Collision response involving rigid objects and involving deformable objects introduce different processes.

A Collision response for rigid objects

Collision response between two rigid object must ensure first that in case of contact, objects do not penetrate each other, and that response forces are properly computed. In case of important relative speed between the two objects in collision, the resulting shock must also be taken into account. This is usually performed by applying an impulse to the objects.

In order to generate forces during contact, several approaches exist. A first method consists in computing a penalty force based on the inter-penetration between the bodies. For each penetrated point, the force can be seen for example as a spring between the surface and the point, aiming at bringing the point back to the surface, and thus make the inter-penetration disappear. This method has been widely used for haptics, and is further explained in Section 3.1.

Another method is to generate a set of constraint forces that ensure the non penetration, as explained in [Witkin et al., 1990]. The forces are computed for each contact points, and solving the created system is seen as a Linear Complementary Problem, which is an optimization problem. If friction forces are added, this system might not have any solution, so a solution is to perform optimization on velocities of surface instead of forces, which guaranties the existence of a solution ([Stewart and Trinkle, 2000]).

Other methods for particle systems directly project the position of the surface particles to the other object's surface.

When the relative velocity between the different objects is important, objects bounce, which creates a discontinuity in the force value. This is performed by directly applying a velocity change, as seen for example in [Hahn, 1988]. The new velocity might be chosen so that the new relative tangential velocity is the opposite to the previous one. Object go away at the same velocity as the one they had before collision, but in opposite direction. It is important to note that this discontinuity makes the

kinematics equation invalid at the time step, so a common approach is to consider the new velocities as initial conditions for a new simulation loop.

B Collision response for deformable objects

For deformable objects, contact is different, since the deformation of the objects caused by the contact must be taken into account. In order to address this, [Duriez et al., 2006] adapted the Signorini's Law, mainly used in offline simulation tools, to the interactive FEM. As depicted in Figure 2.4, we consider the objects D_1 and D_2 in interaction. We consider a point P from D_1, and the corresponding interaction point Q from D_2.

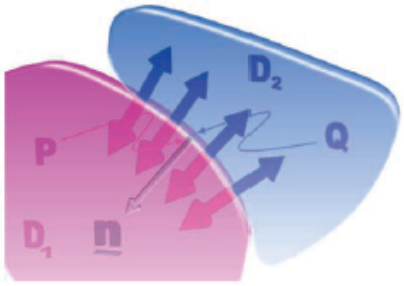


Figure 2.4 – Contact handling between deformable objects D_1 and D_2 ([Duriez et al., 2006])

The normal is called $\mathbf{n}$, with orientation given by QP, and arbitrary direction. The stress at this location follows the following equation:

$$\sigma_n^1(P) + \sigma_n^2(Q) = 0 \tag{2.12}$$

The distance between D_1 and D_2 is given by

$$\delta_n(P) = \langle QP|\mathbf{n}\rangle \tag{2.13}$$

Signorini's Law of contact states that the stress and the minimal distance are linked as follows:

$$0 \leq \delta_n(P) \perp \sigma_n^1(P) \geq 0 \tag{2.14}$$

This means that both quantities are positive, but one must be null, meaning that in order to create pressure, the distance between the objects must be zero (objects have to touch to influence each others). This law only provides information about the normal component of the contact force. In order to get the tangential element of the force, Coulomb's law is used:

$$\mathbf{f_t} = -\mu \parallel \mathbf{f_n} \parallel \mathbf{t} \tag{2.15}$$

with $\mathbf{t}$ the direction of the tangential force.

These equations provide a set of equations to solve, which is performed using a Gauss-Seidel method.

This method has been used by [Otaduy et al., 2009], with implicit constraints in order to ensure non penetration, for the simulation of an important number of deformable objects, ensuring non-autopenetration.

2.2.4 Time Integration

With time dependent equations, time integration allows to determine what the values for the new time step will be, depending on the current state. As previously explained, most of the equations for material simulation are from the following form:

$$\ddot{\mathbf{x}} = F(\dot{\mathbf{x}}, \mathbf{x}, t) \tag{2.16}$$

with F a function. In order to have only first order equations, the equation is rewritten.

$$\dot{\mathbf{x}} = \mathbf{v} \tag{2.17}$$
$$\dot{\mathbf{v}} = F(\mathbf{v}, \mathbf{x}, t) \tag{2.18}$$

A few solution exist to integrate this equation. The explicit (or forward) Euler integration simply transforms the times derivatives by finite difference.

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{v}(t) \tag{2.19}$$
$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t F(\mathbf{v}(t), \mathbf{x}(t), t) \tag{2.20}$$

This model gives explicit values for $\mathbf{x}$ and $\mathbf{v}$, but is stable only for small Δt.
Another method is the implicit (or backward) Euler integration, keeping $t + \Delta t$ on the right side of the equation.

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{v}(t + \Delta t) \tag{2.21}$$
$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \Delta t F(\mathbf{v}(t + \Delta t), \mathbf{x}(t + \Delta t), t) \tag{2.22}$$

This requires more computation, since the system must be solved, but provides it is stable whatever the chosen time step is. Some methods, including [Fierz et al., 2011] and [Fierz et al., 2012] use implicit integration only for ill-shaped elements that threaten stability, with great improvement on computational cost.

2.3 Adaptive Approaches

In order to improve performances, to allow interaction, many methods rely on adaptive representations. As defined in [Manteaux et al., 2016], a model or simulation is called *adaptive* if it automatically adapts the underlying mathematical representation, data structure and/or algorithm at run time, based on the evolving state of the simulated system.

During the simulation, adaptive approaches can be found at different steps. First the geometry of the object is the most intuitive aspect of the simulation that allows for adaptivity. The deformation model itself can rely on different elements, with various associated computation resources. Finally, once the deformation model and desired accuracy has been set, the equation can also be solved adaptively. As described previously, collision detection also heavily rely on adaptive and hierarchical approaches, with the use of Bounding Volume Hierarchies (BVH) for an important simplification of the problem. In this section, we describe the multiscaling approaches on the geometry, simulation and then in the equation resolution.

2.3.1 Multi-scale on Object Geometry

Geometry is one of the aspects of simulation where multiscaling is most relevant. The main idea behind multiscaling on the geometry is that the required geometry precision for an accurate simulation does not need to be the same all over the simulated object. Some parts might need additional precision, which can be provided by local refinement. The main goal behind this is to be able to have a fine simulation, without needing a precise geometry for the entire object, allowing thus better performances.

There are two main approaches to geometric multiscale, mesh multiresolution and embedded models. Mesh multiresolution relies on the coexistence of several resolutions of mesh on an object. The different meshes have the same nature, but the constitutive elements have different sizes. Embedded models rely on an external grid being simulated, and the object being interpolated from the simulation of the overlaying grid.

A Multiresolution Meshes

Multiresolution meshes are amongst the most intuitive multiscale methods in simulation.

When finer elements are locally needed in order to preserve the simulation accuracy, the simulation is performed on finer elements – tetrahedra of smaller size for tetrahedral meshes – in order to get better accuracy, but the rest of the mesh is still simulated using the original coarser elements, for better efficiency. Several aspects need to be handled carefully for this process. First of all, a subdividing criterion is required. A criterion must be chosen to determine when a set of elements are too coarse for the current state of the simulation. Then the subdividing method introduces interesting challenges, such as the preservation of the material properties. Finally, it is important to be able to connect the different resolution to avoid inconsistencies on border elements.

An approach for this is to precompute several different meshes with different resolution for the objects, as proposed by [Debunne et al., 2001]. Since every model is created beforehand, there is no on-the-fly remeshing to perform , and since all these resolutions are coherent models of the same objects, preservation of the material properties is already included in each separate mesh. The two main questions remaining are the criterion choice and the connection between resolutions. Since the chosen deformation model relies on piecewise linear deformation, the quality criterion chosen is the distance from the displacement field to a linear deformation. This difference is strongly linked to the force per volume, thus making the computation easy. In order to link the different models together, they associate different scaled elements using a Voronoï decomposition of the object based on the elements. An element of the finer mesh is linked to one from the parent mesh if it is inside the Voronoï cell of the coarse element. Voronoï decomposition being a partition of space, it ensures unicity of the association. Linear interpolation is then used to represent the object at the border of the multiresolution part. The result can be seen in Figure 2.5.

Other approaches rely on a local remeshing of the object. This approach is quite suitable for mass-spring models, as seen in [Hutchinson et al., 1996], [Ganovelli et al., 2000], [Paloc et al., 2006] or [Zhang et al., 2002].

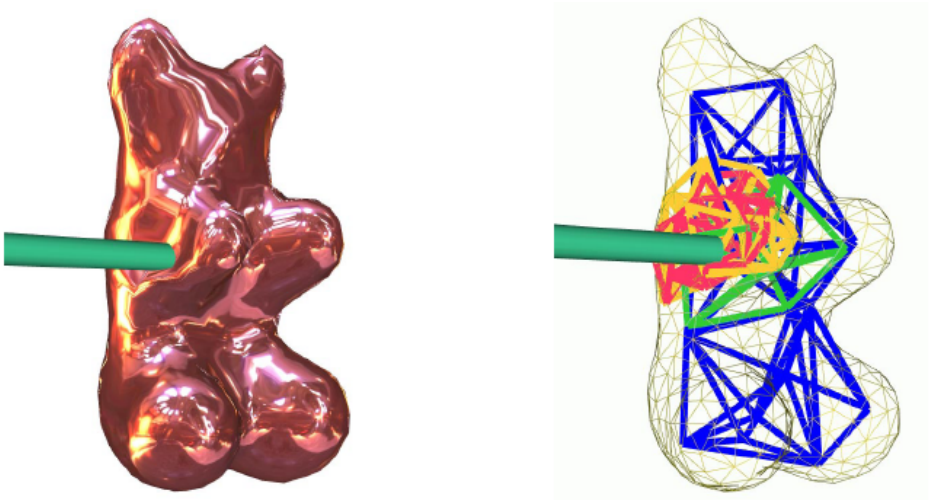


Figure 2.5 – Multiresolution mesh used for fine deformation: the finer mesh is used for deformation where more accuracy is needed ([Debunne et al., 2001])

Early work for this approach has been performed in [Hutchinson et al., 1996], introducing the idea of a local remeshing during the simulation for better performances. It was intended for cloth simulation, in order to locally refine the triangular mesh were important deformation occurred for better precision. This method allowed for a rather coarse base mesh, and the mesh was locally refined at places of important deformation to the finest required resolution. This method would however increase the total mass of the object during the refinement. [Ganovelli et al., 2000], [Paloc et al., 2006] or [Zhang et al., 2002] focused on mesh refinement in the interaction area. If an element lays in the interacting area (with a haptic tool) the mesh is refined. Special care must be taken to set the new values for the masses and the stiffness of the new springs, in order to keep material properties. Computing spring stiffness for the new spring is a much more challenging problem. [Paloc et al., 2006] proposes a solution based on the global material stiffness and the local volumes of surrounding elements, as well as the new spring length.

Local remeshing has also been used for FEM-based simulation, including [Paloc et al., 2006] and [Jun et al., 2006]. At the point of interaction or with elements deforming above a certain threshold, elements are divided. FEM relies on computation of important matrix inverses, and the insertion process must provide efficient methods for the update. [Jun et al., 2006] superimpose a finer mesh to the coarse one at the interaction point, with efficient matrix adaptation, and the overall deformation becomes a linear interpolation from both the coarse and the fine model. This superimposition can be seen as a local remeshing, that can be removed if the region of the mesh does not require additional accuracy anymore at some point of the simulation.

While spring stiffness attribution was a challenge for mass-spring models, a similar problem arises for multi-resolution methods on heterogeneous objects with FEM methods. As explained in [Imai et al., 2013], many multi-resolution methods on heterogeneous objects rely on the fact that the objects is composed of large homogeneous volumes, inside which it is straightforward to attribute elasticity values. In the case of more complex elasticity distributions inside the objects, there is no known

analytical elasticity value to be attributed to coarser elements. In order to address this issue, methods based on inverse dynamics have been proposed. Inverse dynamics computes the deformation of the objects at the fine resolution under a set of external forces. The object at coarse resolution is then deformed under the set of same external forces, and compute the elasticity values that minimize the deformation difference, using numerical optimization. This approach has been performed in [Kharevych et al., 2009], [Imai et al., 2013] and [Chen et al., 2015]. While [Kharevych et al., 2009] provide with a homogeneous coarse mesh, [Imai et al., 2013] create a heterogeneous adaptive mesh and [Chen et al., 2015] create a heterogeneous coarse mesh. However neither of these methods involve haptic, and the effect of such heterogeneity on the perceived force was therefore not investigated.

B Embedded Models

Embedded models rely on a different approach for the simulation. The main idea is to make the physically-based simulation on a grid of voxels and then to interpolate the object inside the grid.

Early work on this domain uses rigid cells for the grid [Botsch et al., 2007]. The object is embedded in a grid of rigid cells, bound together by volumetric elastic energy. When a cell is moved away from another one, the elastic energy associated to the pair of cells is estimated based on the distance between the two cells. Optimization is then used to minimize the elastic energy throughout the object. Adaptive cell size is used for highly deformed regions of the object. In order to get the object geometry, radial basis functions (RBFs) are used for interpolation.

While interesting for shape modeling, this method is not suitable for physically-based simulation, due to its important lack of realism.

For more realism, some methods use an embedded surface. The used mesh is coarse, but in order to simulate fine deformation, a finer surface mesh is embedded, for example in [Sifakis et al., 2007].

However, several challenges are not addressed by the aforementioned methods. For instance, two elements in adjacent cells will be linked throughout the simulation, even if they are completely unrelated inside the object. The interpolation method also does not properly allow for heterogeneous material properties inside the cell. Moreover, the elasticity of each cell should be affected by the empty space contained in it.

A solution to these problems has been proposed in [Nesme et al., 2009] by first redefining the shape functions. The shape functions inside each cell are precomputed in order to have proper interpolation for each node inside the cell, thus allowing the simulation of heterogeneous behavior. In order to be able to have several unrelated parts of the objects in the same cell not affecting each other as a side effect of being extrapolated from the same cell, cell containing several parts of the object are duplicated. One version of the cell is created for each part, allowing for independent simulation. These combined elements allow for heterogeneous branched

objects simulation, such as blood vessels within a liver, as can be seen in Figure 2.6.

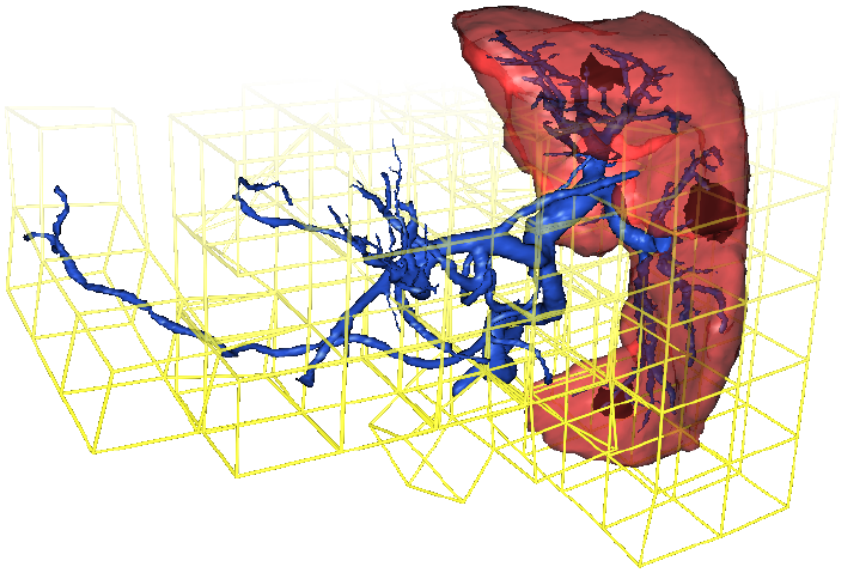


Figure 2.6 – Liver embedded in a coarse grid, with heterogeneous behavior and independent vessel movement taken into account ([Nesme et al., 2009])

As such, the presented model does not really contain many multiresolution features. The size of the grid is for instance fixed. Haptic interaction is also not discussed.

In order to address these challenges, [Tagawa et al., 2013] propose a hybrid solution between multiresolution meshes and embedded models. The object is first represented by a coarse elastic model using the corotational FEM model, in which the heterogeneous properties of the object are embedded. Mesh refinement is possible and simplification if needed are accelerated by matrix change precomputation based on the form of the mesh transformation. Haptic interaction is also addressed. The heterogeneous properties of the liver can be felt through haptic interaction.

2.3.2 Skinning methods

In order to achieve better performances, numerous work has been performed on the simulation itself with multiscale elements. The main idea is that the simulation can be decomposed into simpler steps. It is for instance the case for skinning methods. Skinning methods rely on the simulation of a reduced number of reference frames inside the simulated object, and performing extrapolation in order to simulate the object surface.

Skinning has a long story in character animation. The main idea was first to simulate a skeleton made of rigid bones, and then based on the bones position, create a satisfying surface model. This is illustrated by the work of [Capell et al., 2002]. In their work, they simulate the skeleton, and associate each element of the mesh to a particular bone, in order to derive the deformation of the element from the position

of the bone. The local equations for elastic deformation are linear, allowing for much faster computation. This method is especially relevant for character animation, since most character already have a skeleton, that must be taken into account for animation. Results of this method can be seen in Figure 2.7.

Figure 2.7 – An underlying skeleton is used for character animation ([Capell et al., 2002])

While quite suitable for character animation, rigid bones introduce a too hard constraint for more general object simulation. This idea has thus be extended with the use of complete deformable bones. The object is deformed through the deformation of control frames, and the remaining of the object is simulated by skinning.

With such methods, it is possible to simulate objects without obvious skeleton structure, such as cloth. The control frames are then chosen based on the required precision, and not on an actual skeleton anymore.

With this method, it is possible to simulate objects as unstructured as cloth. In order to achieve this, [Kavan et al., 2010] notice that the usual method skinning method for character animation gives good results for cloth animation, and propose a method for efficient parameters evaluation. They perform optimization for all different parameters within the skinning process, such as bone transformations, vertex weights and rest pose position. With careful simplification of the dimensionality of the optimization problem, good animation results can be obtained through reasonable preprocessing time.

While this method works fine for homogeneous objects, some work has been performed to be able to animate heterogeneous objects as well. Taking the heterogeneous aspect of the material would necessitate the creation of an important number of bones, which is not compatible with the interactivity required by the applications. An approach by [Gilles et al., 2013] to address this problem is to take the compliance, ie the ability to change volume under a given pressure, of the object into account for the shape functions, while still relying on a limited number of bones. The idea is to weight the distance to the frames by the compliance when computing the shape function, and not the euclidean distance. With this method, heterogeneous properties can be simulated.

For a more extensive presentation of all skinning methods and bone choice, please

refer to [Di Giacomo et al., 2007] and [Jacobson et al., 2014].

2.3.3 Equations with Different Solving Levels

Equations resolution can be implemented using a adaptive approach. This is the case in [Edwards and Bridson, 2014], where they consider the equation solving as a Galerkin problem. In order to find the solution to a system of equations with a potential infinite dimensioned solution, a Galerkin method finds an approximate solution of the projection on a finite space. The found solution is identical to the solution on this finite space, but some error remains in the orthogonal space to the chosen solution space. Each new dimension added to the solution space grants a better solution. [Edwards and Bridson, 2014] uses this property for fluid simulation. The simulation is performed on a coarse simulation grid, and is then interpolated inside the cells. The dimension of the solution space depends on the precision needed at this location. For instance, elements near the surface are simulated using a greater number of dimensions than the elements interior to the liquid, as depicted in Figure 2.8. Elements interior to the liquid have a solution space of dimension 1, ie linear interpolation.

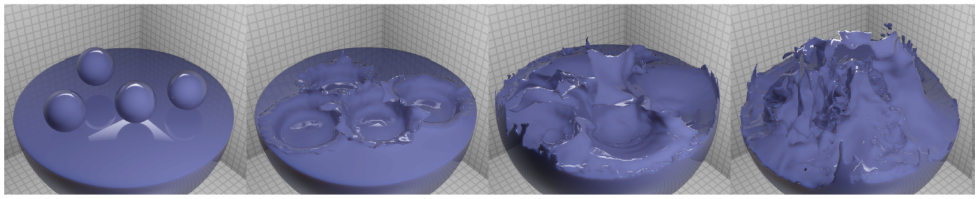


Figure 2.8 – Fluid simulation using a greater solution space for surface elements ([Edwards and Bridson, 2014])

2.4 Conclusion

In this section, we have described different aspects of the physically-based simulation of deformable objects. Many adaptive methods have been proposed in order to improve simulation performances, with a locally simplified – or refined – model to allocate resources only where they would be best exploited.

These methods have quite distinct application fields. Initially quite limited to the simulation of characters with rigid skeletons, skinning methods have been used with success to simulate deformable unstructured objects, but they rely on an initial skeleton choice, which limits the level of adaptivity, and topology changes.

Embedded models offer more flexibility regarding the adaptivity, and are quite efficient, due to the simplicity of the model that they rely on. As with skinning methods however, this method is not suited for topology changes, and the accuracy is limited to the precision of the overlaying model.

Adapting the solving level of the equations offers great flexibility and enables to set the desired accuracy in advance, but it is not clear how this method can be used with arbitrary simulation methods.

Multiresolution meshes offer great advantages, because they can provide an important number of simultaneous interaction scales, since several resolutions can be

used for the same object. They also provide a good realism, because the simulation method is performed on a model of the object, with physically-based simulation. Bringing more accuracy at some point of the object does not remove any accuracy for the rest of the object. However, this method can under special circumstances undergo severe performance problems. If an important deformation is applied to the entire object, such as overall torsion, the object would have to be completely refined (not only locally), and the performance would drop.

While providing with a more efficient simulation, the effect of these simplifications on perception, especially haptic, is often not considered, especially in the case of heterogeneous objects, and methods taking perception into account would provide new validation means.

3 Haptics

3.1 Introduction

The previous section described how to simulate a virtual environment. This section presents related work to the haptic interaction with such a virtual environment. Haptic feedback is the information provided through the sense of touch to the user. There are two kinds of haptic feedback. First the tactile feedback, providing information about the surface of the object, and then the kinesthetic feedback, providing information about the force applied to an object.

The force that can be applied on a deformable object has different components, called Degree-of-Freedom (DoF). Each translation or rotation that can be applied on the object is a degree-of-freedom. Without any constraints, an object has six DoF, three translations and three rotations.

Interaction with an object requires interfaces. For haptic feedback, we need a material device. The user moves the interface, and be provided with haptic feedback depending on the scene configuration and on the action performed on the virtual environment. The number of DoF available for the interaction strongly depends on the chosen device. For example, the devices in Figure 2.9 have all six DoF for positioning the object, but only the Virtuose (Haption, Soulège sur Ouette, France) has six DoF of force feedback. The Geomagic Touch (3D Systems, Rock Hill, USA) and the Falcon (Novint, Washington, USA) have respectively three and up to five DoF for feedback.

(a) **(b)** **(c)**

Figure 2.9 – Different haptic devices: (a) Geomagic Touch, (b) Falcon and (c) Virtuose

For a full haptic interaction, several aspects must be taken into account. It is

first important to create a link between the physical haptic device and the virtual environment. Defining how the movements of the interface impact the virtual environment and reciprocally how the dynamics of the virtual environments impacts the interface are important elements to deal with. Part of this problem is the definition of the haptic coupling scheme. Since a major part of the interaction comes from the contact between several objects, collision detection and collision response are also key components. Haptic interfaces require high refreshment rate in order to be perceived as correct by the user. Frequencies up to 1KHz for collision between rigid objects are estimated as correct. Such frequencies, much higher than the 30-200Hz required for visual display introduce new challenges for haptic compared to only physically-based simulation with visual feedback. Stability is also a challenge when it comes to haptic interfaces, since unwanted instabilities can lead to important drop in plausibility, and is not a trivial problem, since the unpredictable behavior of the user must be taken into account. The fact that all the process is not continuous but relying on fixed time step can also lead to some major stability problems. A criterion for this is that the haptic device must be passive, ie not introducing any energy in the system, no matter what the circumstances.

A complete survey on most challenges related to haptics can be found in [Lin and Otaduy, 2008]. We hereafter develop some of the key aforementioned aspects.

The stability problem is traditionally associated to the problem of the virtual wall [Colgate et al., 1995]. The goal is to design the contact between a virtual tool and a virtual wall, and to design the collision between the two to be as stiff as possible. Most of the stability concerns can be modeled as virtual wall problems. The formulation allows for explicit formulation of the passivity condition, and to determine under what circumstances a setup can be considered passive. One of the elements on which it is possible to act to make the setup passive is the damping of the system.

The problem of the virtual wall also raises the question of contact handling between the haptic device and the environment. Due to the discontinuity of the simulation, the position of the haptic device is likely to intersect objects, and to penetrate it. Force computation was performed based on the distance to the surface, leading to undesirable effects near sharp edges. The God-object Method proposed by [Zilles and Salisbury, 1995] considers an intermediate representation of the object that would not penetrate the object. During contact, the god object would remain on the surface, and the returned force is computed based on the distance between the god object position and the interface position. Computing the position of the god object on the surface can be done through the computation of a set of constraints representing the local geometry of the interacting object. It can ensure non penetration inside the object.

This method has been extended by [Ruspini et al., 1997] with the notion of virtual proxy. The principle of having an intermediate representation is kept from the previous method, but in addition, contact forces are better handled, with for example force shading in order to smooth the force changes during a tactile exploration. The movements of the proxy on the surface are also taken into account in order to simulate phenomena such as friction. All these are the key elements for haptic interaction.

3.2 Adaptive Acceleration Methods for Haptics

3.2.1 Online Geometry Adaptation

Similarly to what is performed for physically-based simulation, haptic interaction often relies on adaptive models in order to get better performances. Geometry adaptation is an intuitive way to get better local accuracy and is suitable for haptic interaction as well. Part of the haptic specificities are the remeshing criteria, both for the decision to remesh and for the part of the object to remesh. Concerning the area to remesh, the remeshing usually relies on proximity with the interaction tool, with an arbitrary distance to the tool. The decision to remesh depends on the application.

For instance, for tactile feedback, the remeshing can be performed based on the velocity of the finger, as proposed by [Kolcárek and Sochor, 2004]. The main idea is that if the finger moves quickly on a surface, less details should be perceived. This allows for simpler collision detection and response, and thus better performances, but does not significantly hinders the perception of the surface, since the fingertip naturally perceives less details when exploring surfaces more quickly. The mesh adaptation is performed through continuous LoD, with recursive vertex merging. This allows a fine resolution choice, but can lead to local problems, since two merging vertices can create an edge inside the fingertip. The speed required for mesh simplification is chosen based on a linear function of the velocity. The result for this method can be seen in Figure 2.10.

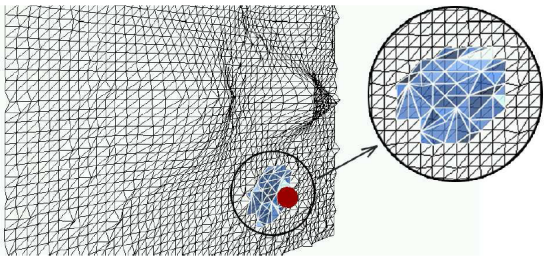


Figure 2.10 – Velocity-driven LoD: the mesh is coarsened when the velocity of the finger is high enough ([Kolcárek and Sochor, 2004])

This method provides a remeshing criterion for tactile interaction, but is not suitable for kinesthetic interaction. While tactile interaction would gain benefit from a local coarsening of the mesh for faster collision response, the symmetric approach is usually chosen for kinesthetic interaction, having a coarse model with local refinement for better local accuracy in the force computation. This approach is especially relevant with the interaction with vertex manipulations on deformable objects. Several criteria can then be chosen for mesh subdivision.

A first approach taken by [Tagawa et al., 2012] is to consider a physically-based criterion, the local deformation, as the criterion for subdivision. A high deformation signifies that an important deformation is being performed in the neighboring area, and more precision can thus bee needed. Subdivision if the the mesh elements can allow for local better precision, as well as better global mesh quality. This method is interesting, because can enforce quality criteria on the object while controlling the

size of the elements. It also does not depend on an arbitrary zone defined by the user. However, the perceptual necessity for refinement of the mesh is not discussed.

For applications where the user's perception is a major factor, a criterion based on the user's perception might also be relevant. This approach has been chosen by [Payandeh et al., 2005]. They investigated the user's perception for the interaction with both rigid and deformable objects. In both cases, they investigated what was the coarsest possible mesh they could use without the user feeling the difference with as fine model. Interaction with the rigid object was tactile whereas interaction with the deformable object was like the previous method, ie vertex manipulation. In order to determine what resolution was suitable for their purpose, they have conducted perceptual user studies.

In both cases, the operation mode was similar. Users were asked to interact with the object at finest resolution and at coarser resolution, with several different possible coarse resolution, also described in [Zhang et al., 2003]. The haptic device used for this test was a Phantom device, with 3 DoF feedback. For each pair of objects, they were asked whether they had perceived a difference in the model or not. The visual model was made such as it would not make any visual difference between different resolution of haptic models, as depicted in Figure 2.11. This method gives the suitable JND for the interaction scheme. The refinement zone was however arbitrarily chosen.

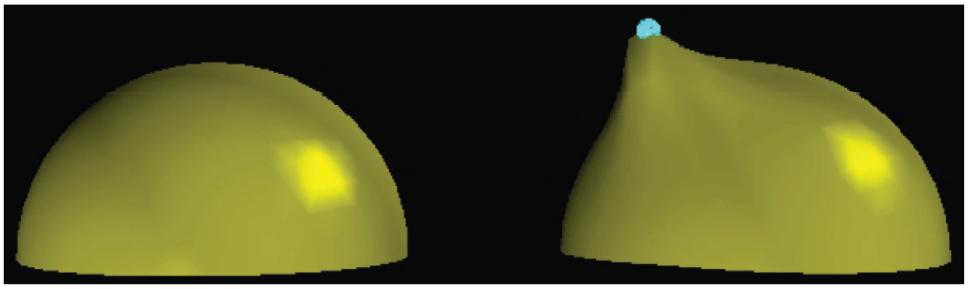


Figure 2.11 – A perceptual criterion is chosen for mesh refinement: the criterion for mesh refinement is chosen using perception data from a user experiment ([Payandeh et al., 2005])

3.2.2 Multirate Methods Using Models Juxtaposition

As stated in the introduction, the refreshment rate needed for haptic interaction is much higher that the one required for the rest of the simulation, by a factor up to about 20. On the other hand, features required for proper haptic rendering are not as precise as those required for acceptable visual feedback. Based on these observations, several methods have been developed using at least two different models, one precise model for the actual physically-based simulation, and a simplified one for haptic interaction. The idea is that a simpler model requires less computation and allows thus smaller time steps. The approximations made for the haptic model often rely on a simplification for the collision response. For instance, [Mendoza and Laugier, 2000] attaches a virtual bar to the haptic object directed in the object's movement. If the bar touches objects (that can be deformable) in the scene, the actual collision detection and collision response is

performed with neighboring elements of the touched object. In order to handle collision response, a proxy representation is used.

This idea of intermediate representation for collision response can also be found in [Davanne et al., 2002]. In order to handle collision detection for haptic interaction, the scene is decomposed into voxels, for first pruning of pairs of potentially colliding objects. The object is also represented as a set of spheres, based on which the collision response can be computed. Careful note of the direction of the objects in the voxel representation can allow an even more efficient pruning phase.

While these two methods relied on an intermediate representation for collision response purposes, some other methods rely on the different in haptic rate required based on the stiffness of the objects in order to have the best possible performances. Intrinsically, some objects in the scene might require higher computation frequencies due to their material properties. While a large deformable object might not need high performances, stiff objects need frequencies comparable to those for the virtual wall problem. [Peterlik et al., 2011] consider objects that might need faster simulation to be curved beam-like objects, such as depicted in Figure 2.12. This difference in simulation rate induces a total asynchronous simulation. Under these conditions, collision detection and collision response must be handled carefully, since contrary to the previous methods, the collision models for each object are computed at different rates. The models used for collision are taken from last simulation step for each object. The contact constraints are then computed using the free positions of the objects. These constraints are then further synchronized with each object.

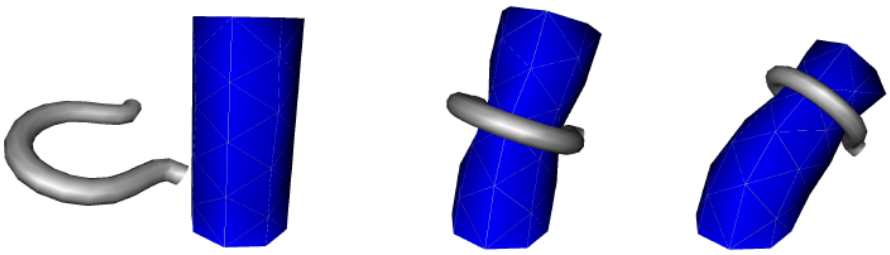


Figure 2.12 – A stiff clip is simulated at higher rates than the deformable cylinder ([Peterlik et al., 2011])

3.2.3 Multiple Objects Interaction

In many cases, haptic scenes must handle an important number of elements. In order to still be able to compute feedback efficiently, many models have been proposed to simplify the interaction between objects. One way to perform this combines a certain number of aforementioned methods. The idea of [Otaduy and Lin, 2005a] is to use a multiresolution model for collision detection. This representation decomposes the object into a set of convex parts. This representation has been chosen due to the computational efficiency of convex hull-based collision detection algorithms. In order to have proper multiresolution possibilities, they introduce a method allowing to merge adjacent convex parts into a new bigger convex part. The method merges pairs of edges

in order to obtain a coarser part. The new created vertex is then relaxed to its final position based on its connectivity. After that step, a convexity test is performed in order to check whether the edge collapsing is valid or not. An example of the obtained multiresolution can be seen in Figure 2.13.

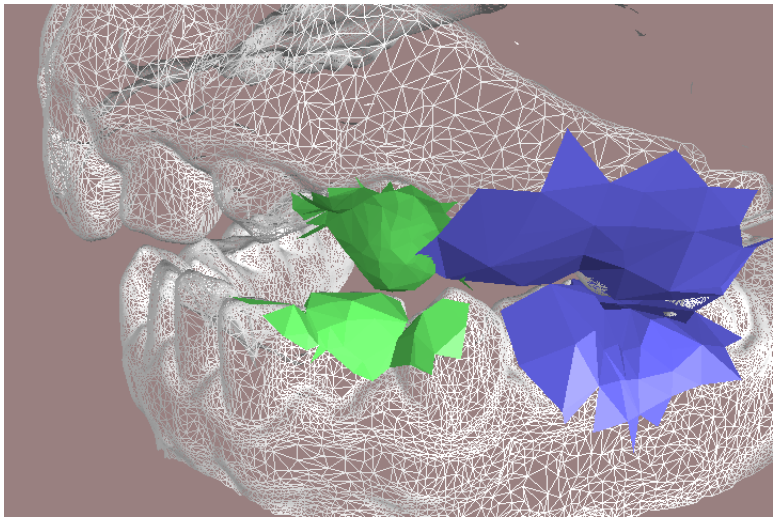


Figure 2.13 – Two jaws interacting with a multiresolution model ([Otaduy and Lin, 2005a])

The chosen strategy for collision detection is to use the coarse models, and then to refine when a quality criterion is not verified. The quality criterion represents the fact that a surface detail has been missed, but should have been felt, ie if the lack of local surface detail is perceptible. The metrics used for this criterion is a surface deviation using the area of contact and the resolution. If the deviation is too important, the mesh needs to be refined.

While this methods focuses on a simplification of the collision model for the interaction, several other methods act on the number of objects considered in interaction for collision response purposes. If a scene contains an important number of objects, computing the collision response accordingly to every other object can prove to be too costly, and incompatible with interactive applications. Some methods reduce the number of objects that need to be taken into account.

A method uses contact information from the collision detection to build a sub world dedicated to haptic rendering [Glondu et al., 2010].

A contact graph is created in order to know what are the objects in contact and in indirect contact to the proxy. Since contact can appear between successive time steps, objects close enough from the contact objects are also taken into account. If too many object are in indirect contact to the proxy, a limited amount of objects are taken into account, and energy-based boundary conditions are created.

Other methods, such as [Susa et al., 2011] consider objects within a fixed distance to the proxy. In both cases, the local simulation rate is much higher for the rest of the simulation. In order to have consistency between the local representation and the

rest of the simulation, [Susa et al., 2011] compute the accelerance of the objects in the global scene, and include it in the local simulation.

These two methods isolate a part of the scene considered more important for haptic rendering, in order to simulate it at a higher rate. Boundary conditions must be respected in order to keep consistency between the local haptic world and the global scene. They are however limited to rigid objects.

Some methods also exist for deformable objects. While many applications have studied the interaction between a rigid tool and deformable objects, for example for surgery simulation (see [Otaduy and Gross, 2007] or [Mahvash and Okamura, 2005]), few methods exist with only deformable objects interacting. The method proposed by [Garre and Otaduy, 2009] allows the user to grasp the deformable object with a low dimensional haptic device. The proxy for interaction is a part of the deformable object. The reason for that approximation is that a natural grasp would reduce the deformation of the grasped zone, and it also allows the displacement of the entire grasped zone with a 6 DoF input. Two simulation threads are used, a visual thread and a haptic thread, with different simulation rate, as discussed before. In order to get proper force information during the simulation, between two visual time steps, the resulting force (both internal and external forces) applied on the grasped zone is evaluated at each visual time step with a defined set of constraints. This coupling force is then linearized, similarly to what is performed in [Otaduy and Lin, 2005b]. For proper force within the haptic thread, the force is evaluated with current parameters.

3.3 Conclusion

In this section, we have presented the related work to adaptive methods for the haptic rendering of complex environments.

First, acting on the simulation framerate, many methods use different models to separate the visual feedback and the haptic feedback, relying on fast update methods for haptic feedback. These methods also allows to attribute different framerates based on the physical properties of the object. It should however be handled cautiously, as to not create inconsistencies between the different models.

The geometry of the object can also be adaptively simplified, and this change can be performed in a way that is not perceived by the user. These methods were however used on a simple geometry, with a simple deformation behavior, which limits the potential use. Overall, these methods have a great potential for the efficient interaction with complex environments, but would require specific approaches in order to handle this complexity.

4 Conclusion

In this chapter, we have presented results associated with the three major components involved in an haptic interaction loop with a virtual environment, namely the perception of the user of the interaction, the physically-based simulation of the environment, and the haptic rendering of this interaction. We have first shown that the haptic perception of an environment suffers from limitations, with the incapacity to discriminate between

two stimuli if there are within stimuli-dependent boundaries. Haptic perception has also been shown to be greatly influenced by other perceptual modalities, and above all the visual modality. We then presented results associated to physically-based simulation and haptics, and focused on adaptive methods, that dynamically change the resource allocation in the virtual environment. These methods need a criterion in order to decide where and when the resources are most needed, and we believe that the perception inaccuracy represents a natural leeway that allows adaptive methods to be most effective, with a limited impact on the perceived interaction with the virtual environment. Conversely, the influence of the visual modality over haptic perception must be accounted for, and the emergence of new display technologies calls for careful evaluation of the influence that these displays can have on the haptic perception, which must be the first step for any application that aims for accurate haptic feedback.

Elasticity-based Clustering for Haptic Interaction with Heterogeneous Objects

3

Contents

1 Introduction

As almost all materials of our daily life are non-homogeneous, the ability of interacting with such materials is crucial when designing physically-based simulations of virtual environments. The simulation of non-homogeneous objects remains however computationally demanding and often leads to simplification in the material simulation. Approaches such as embedded models [Nesme et al., 2009] or frame-based methods

[Gilles et al., 2013] for instance have been proposed to address the heterogeneity of the object.

The computation time performance issue associated to the physically-based simulation and the haptic interaction with complex deformable heterogeneous objects is a challenge that needs to be addressed. Focusing on the physically-based simulation aspect, many methods rely on a geometric multi-resolution of the object in order to simulate deformable objects in an efficient manner. However, performing geometric multi-resolution on heterogeneous objects raises the issue of elasticity attribution at low resolution, as discussed in section 2.3.1. An important factor is that this geometric multi-resolution method should not have a too dire effect on the haptic interaction with the object.

In this chapter, we propose a novel approach that facilitates the computation of elasticity parameters of multi-resolution methods in the context of haptic interaction with a heterogeneous deformable object. Our method creates clustered volumes with homogeneous elasticity inside the object, while focusing on the distribution pattern of elasticity inside the object and its effect on haptic perception. Our method coarsens the heterogeneity of our object, while keeping important features of the elasticity of the object. In particular, the overall elasticity of the simulated object remains similar. We also propose an evaluation of our method using a perception-based criterion, in order to estimate the perceptual proximity between the initial complex object and the object coarsened with our method.

In the remainder of this chapter, we first present the overall pipeline of our method with its four main phases in Section 2. The evaluation protocol of the method using a perception-based quality criterion and the corresponding perceptual results and performance gains are shown in Section 3. A use case illustrating our method, as well as the haptic rendering scheme are then presented in Section 4. The chapter concludes with a discussion on the method, and possible future work.

2 Description of our approach

2.1 Method overview

As explained in [Imai et al., 2013], most methods using multiresolution on heterogeneous objects make the assumption that the object is separated into homogeneous domains. However, objects with complex elasticity distribution cannot be as easily separated into clear homogeneous domains. Such objects require methods that take into account the heterogeneity of the object, whatever the initial distribution of elasticity is within the object. The coarsening of the elasticity of an object should be carefully handled as it could greatly influence the haptic rendering. We propose a method that aims at clustering the elasticity of an object while keeping similar haptic perception for the user. This method is particularly well suited for coarse interaction with the deformable object, such as probing with a large tool, or having two deformable objects pressed against each other. Our method is based on the use of co-rotational FEM [Müller and Gross, 2004] and allows the setting of elasticity values of a coarse mesh based on the elasticity distribution of a finer mesh, as illustrated in Figure 3.1. It

is important to note that this method consists in a simplification of the object geometry and elasticity distribution, but does not depend on the chosen simulation method.

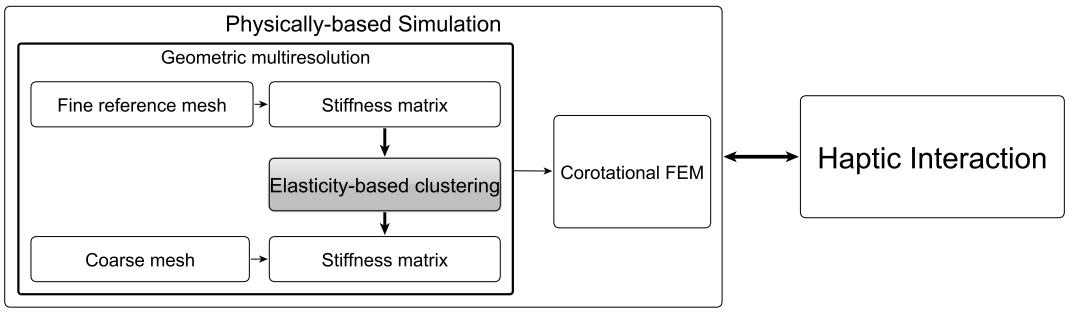


Figure 3.1 – Simulation pipeline with elasticity-based clustering: our novel method (in grey) simplifies the elasticity distribution inside a fine mesh in order to compute the elasticity inside the corresponding coarse mesh.

2.2 Our Algorithm: Elasticity-based Clustering

We introduce a novel approach called "*Elasticity-based Clustering*". The main principle is to modify the elasticity distribution of an object in a way that creates large homogeneous domains inside the object, while preserving the resulting elastic feedback force, as illustrated in Figure 3.2. Our method comprises 4 main phases:

1. Initialization
2. Homogenization
3. Aggregation
4. Geometric Coarsening

In the Initialization phase, the goal is to regroup tetrahedra of similar elasticity, in order to create important homogeneous regions inside the object, for an easier geometry coarsening. The first step consists in defining similar elasticities through elasticity binning. Tetrahedra with similar elasticity are then clusterized. The Homogenization phase attributes an average elasticity value to elements of similar elasticities, reducing the number of different elasticities inside the object. The Aggregation phase modifies the elasticity of the tetrahedra, in order to reflect the clusters, creating more important homogeneous volumes. Last, the Geometric Coarsening phase attributes elasticity values to the tetrahedra from the coarse mesh. The following sections describe the aforementioned phases.

2.2.1 Initialization

The input of our method is a volumetric tetrahedral mesh representing the object, as

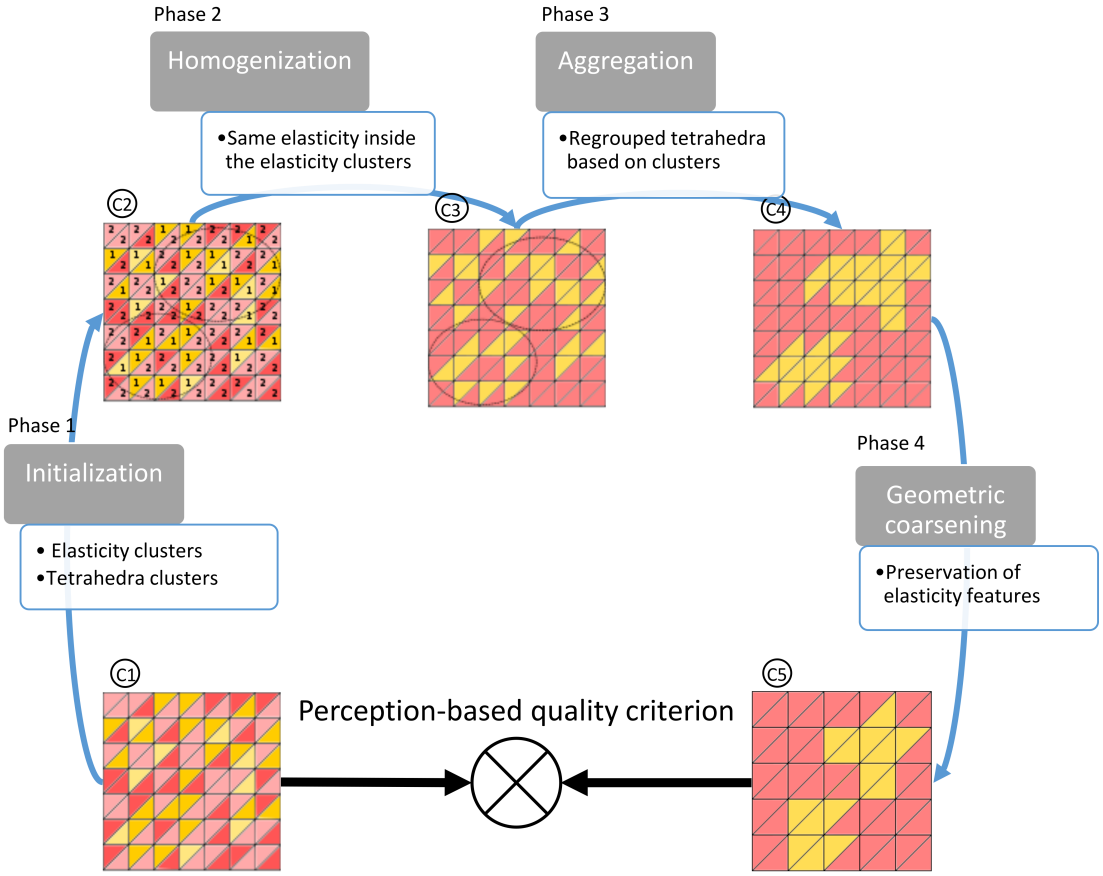


Figure 3.2 – Schematic diagram of our elasticity-based clustering approach. It is composed of 4 phases: 1) Initialization: creates elasticity bins and tetrahedra clusters inside these elasticity bins. 2) Homogenization: unifies the elasticity inside the elasticity bins. 3) Aggregation: modifies the elasticity inside the object according to the tetrahedra clusters. 4) Geometric coarsening: attributes elasticity to the coarse mesh. Cube 1 (C1) represents the initial distribution of the elasticity. Red and yellow elements represent respectively stiff and soft elements, and the different tints for each color represent close elasticity values. Cube 2 (C2) represents the cube after the initialization phase, with information added to the tetrahedra: all tetrahedra are labelled with their elasticity bins (1 or 2); and barycentric clusters have been computed for all elasticity bins except one (here two clusters for the yellow elements). After the homogenization phase, in Cube 3 (C3), all tints for each color have been modified to the homogenized elasticity value for each elasticity bin. Tetrahedra are then regrouped for each cluster around the cluster barycenter in Cube 4 (C4), creating two large homogeneous volumes. A coarser mesh is finally used for Cube 5, with fewer elements, and the elasticities are chosen to match at best the elasticity from the fine resolution.

illustrated in Figure 3.2, Cube 1. This object is heterogeneous, which means that to each tetrahedron t_i is associated a local Young Modulus $E(t_i)$. The goal of the method is to reorganize the elasticity inside the object in order to simplify its structure and to create large homogeneous volumes inside the object for more efficient simulation. The first step of this simplification is to reduce the number of different elasticities inside the object, by regrouping tetrahedra of similar elasticity together, for a later homogenization of similar tetrahedra. This process is similar to the data analysis method called *binning*, in which data is partitioned into a relative smaller number of categories based on similarity values, for easier analysis. We here adapt this method for the elasticity value of the tetrahedra.

In order to achieve this, the main problematics is to set a number of bins, and the elasticity values associated to each bin. There are many possible ways to attribute elasticity values to each bin, with different advantages. Some possible methods include:

- Once the number n of different bins has been (arbitrarily) chosen, sort the elasticity values of each element in growing order, and divide the created element list into n bins with equal numbers of elements. This methods has the advantage to have bins of equal size, which might reduce the computation cost of subsequent treatment, but makes it possible to have in each bin elements with great elasticity difference;

- Use a data classification method on the different elasticity values, such as a k-means clustering method, in order to automatically detect elements of close elasticity values.

- Divide the elasticity range into a set of equal size bins.

Our method does not depend on a specific elasticity attribution method. We decided to use the later method, ie to divide the elasticity range of the tetrahedra ($E_{max} - E_{min}$) into $numberElasticity$ elasticity ranges (later referred as *elasticity bins*) of same size $E_{max} - E_{min}$ $numberElasticity$. This method enforces an elasticity similarity between elements from the same elasticity bin. While $numberElasticity = 1$ represents the most extreme possible simplification, i.e. completely homogenizing the object, a too large number does not simplify the object enough, and a compromise between efficiency and accuracy must be found. To each tetrahedron is then associated an elasticity bin. It is important to note that, even if the elasticity ranges are of same size, the number of tetrahedra is not necessarily similar in each elasticity bin.

The second step for mesh simplification is to regroup tetrahedra of similar elasticity together, in order to create important homogeneous volumes inside the object. To do so, the tetrahedra inside each elasticity bin are clusterized using a spatial criterion, with standard *k-means* clustering using an Euclidean distance. After the initialization, each tetrahedron has then an elasticity bin and belongs to a cluster of close tetrahedra within its elasticity bin. The result of this step can be seen in Figure 3.2, Cube 2.

This phase requires important computation time, up to a few seconds with an important number of elasticities and clusters on a large mesh, but is performed entirely offline. It is also important to note that in this phase, no modification to the object is performed. The object is only given more information required for an online execution

of the elasticity simplification. Actual modifications to the elasticity are described in the two further phases of the algorithm.

2.2.2 Homogenization

In this phase, each tetrahedron is in a bin of elements with similar elasticity. The next step is then to homogenize these tetrahedra, by giving them the same elasticity. Several approaches can be used to address this homogenization step:

- Choosing the median among the elasticity values of the elements inside the elasticity bin. This method has the advantage to be quite robust towards a few outlier elasticity values that might be present in the elasticity bin, but does not take into account the volume of the elements in the elasticity bin;
- Choosing a mean value weighted by the volume of the elements in the bin.

We chose the later approach. The mean elasticity is computed as follow, for each elasticity bin EB:

$$E_{mean} = \frac{\sum_{t \in EB} V_t * E(t)}{\sum_{t \in EB} V_t} \tag{3.1}$$

This value for the elasticity corresponds to the resulting elasticity value for two beams in parallel and is similar to the computation performed in [Nesme et al., 2006]. While [Kharevych et al., 2009] have shown that this method is not accurate for the homogenization of elements with important stiffness difference, this averaging is performed with elements of close elasticity, and gives a force-wise proper approximation. Figure 3.2, Cube 3 shows the results of this phase.

2.2.3 Aggregation

In this phase, the number of different elasticities taken into account is already reduced, from $numberTetrahedra$ to $numberElasticity$. The next step of simplification for the object is then to regroup tetrahedra with same elasticity. The tetrahedra have already been regrouped inside clusters during the Initialization phase. The only remaining step is thus to aggregate them together, in order to create important homogeneous volumes inside the object. To do so, the tetrahedra inside the object are not actually displaced, but the elasticity of each tetrahedron is modified to build the clusters. This process is only a displacement of elasticity values and does not modify the geometry of the object, which would be too costly and would not be meaningful in our context. It is also important to note that the overall volume for each elasticity remains the same, in order to preserve an overall equal elasticity.

The process is described in Algorithm 1: each elasticity bin is aggregated in turn, meaning that clusters are aggregated inside each elasticity bin. In order to keep the overall elasticity similar, the volume of modified tetrahedra for each elasticity E_{bin} must be equal to the volume of tetrahedra with elasticity E_{bin} before the aggregation process, as depicted in Figure 3.3. The volume of modified tetrahedra is recorded in order to enforce this condition.

Algorithm 1: Aggregation of elasticity bins

```
forall the ElasticityBin do
    forall the TetrahedronCluster do
        ModifiedVolume = 0;
        Tetrahedron center = getTetrahedron(ClusterBarycenter);
        TetrahedronList Interestlist = List(center);
        while ModifiedVolume < ClusterVolume do
            Tetrahedron interest = InterestList.pop();
            for Tetrahedron neighbour ∈ interest.NeighbourTetrahedra do
                ComputeInterest(neighbour);
                InterestList.add(neighbour); (Sorted by measure of interest)
            if NotAlreadyModified(interest) then
                interest.ModifyElasticity(ElasticityBin.Elasticity);
                ModifiedVolume += interest.Volume;
                NotAlreadyModified(interest) = false;
```

In order to preserve notable features of the original distribution of elasticity, several properties need to be considered.

- The barycenter of the elasticity cluster in original and aggregated states should be at the same place, in order to preserve the locality of elements with this elasticity. The goal of this method being to create large homogeneous volumes, the aggregated cluster should have elements grouped around the barycenter, and not create thin parts by straying too far away from the barycenter.
- The distribution of the elasticity around this barycenter in the aggregated version should also correspond to the distribution in the original version of the elasticity.

As depicted in Figure 3.3, with original elements around the barycenter, the star-shaped aggregated volume is large enough for geometric coarsening, while keeping the distribution pattern of the aggregated elements. In order to achieve the aggregation, the elasticity is incrementally modified, starting at the barycenter of the cluster. For this purpose, a measure of interest is created for a tetrahedron, based on the distance to the cluster barycenter and on the relative orientation between the tetrahedron to be modified and the original tetrahedron w.r.t the cluster barycenter. The final measure of interest is a weighted sum of the distance factor and the orientation factor. The interest measure for a tetrahedron t adjacent to the current cluster C, taking into account the position of the position of a tetrahedron t_{init} to be aggregated, with barycenters $\mathbf{b}_t$, $\mathbf{b}_C$ and $\mathbf{b}_{init}$ is

$$\text{interest\_measure} = \alpha \times \text{distance\_factor} + \beta \times \text{orientation\_factor} \quad (3.2)$$

$$\text{with distance\_factor} = \|\mathbf{b}_t - \mathbf{b}_C\| \quad (3.3)$$

$$\text{and orientation\_factor} = \frac{|\langle \mathbf{b}_t | \mathbf{b}_{init} \rangle|}{\|\mathbf{b}_t\| \times \|\mathbf{b}_{init}\|} \quad (3.4)$$

The tetrahedra are incrementally modified, starting at the center of the barycenter according to this measure of interest. A tetrahedron which elasticity has been modified can no longer be modified. It is important to note that the proximity to the barycenter and the respect to the original distribution properties are not explicitly enforced, but rather derive from the definition of the interest measure. A stronger focus on either property can be performed by adjusting the weighting factors in the definition of the interest measure. The weighting factors must also take the size of the objects into account, in order to give relevant importance to the distance factor.

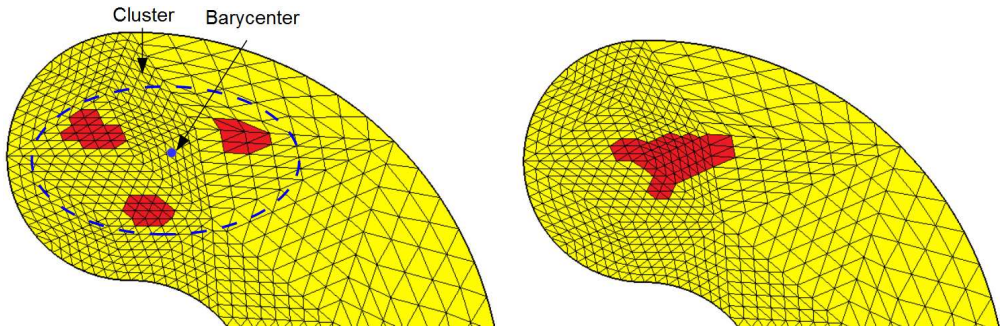


Figure 3.3 – Example of clustering: our algorithm transforms the elasticity of mesh elements: from 3 sub parts of same elasticity (left image), there is only one homogeneous sub part after the end of the algorithm (right image). The aggregated part has similar volume compared to previous sub parts, and keeps distribution features from the previous position of the sub parts.

Since each tetrahedron can only be modified once, the last elasticity bin to be considered can only modify the tetrahedra that have not been modified by any previous elasticity bin. The location of the tetrahedra might therefore not depend on the distribution of the tetrahedra from this cluster in the object. This raises the fact that the order of aggregated elasticity bins has a non-negligible impact on the final configuration.

This entire process can be performed multiple times, with decreasing number of elasticities and of clusters, for further simplification. The number of clusters represents a trade-off between the presence of larger volume, and thus possibility of important coarsening, and similarity to the original elasticity distribution. The level of accuracy of our method is thus mainly decided by the chosen number of clusters. Figure 3.2, Cube 4 shows the aggregated cube, with only 2 large homogeneous volumes.

2.2.4 Geometric coarsening

The last aspect of this method is to attribute elasticities to the elements of coarser resolution meshes. The object is already decomposed into large homogeneous volumes, so the elasticity of the coarse elements inside these volumes is straightforwardly set to the volume elasticity. At the limit of these volumes however, the same procedure cannot be used. Existing methods compute the new mesh based on the elasticity in order to keep homogeneous regions in the coarser mesh, requiring either a computation on-the-fly of the new geometry or a precomputation based on a previous known elasticity

pattern. These methods are not adapted to an arbitrary elasticity distribution inside the mesh.

A first precomputation phase computes the correspondence between the tetrahedra from each mesh resolution. In order to determine the elasticity of coarser elements, the elasticity of the finer elements located at the same place of the object must be known. This precomputation computes the correspondence C_{ij} for each pair of tetrahedra T_i and T_j respectively taken from the fine and the coarse tetrahedra:

$$C_{ij} = \frac{volume(T_i \cap T_j)}{volume(Ti)} \tag{3.5}$$

This step provides a comparison matrix between the two meshes, that can be arbitrary meshes sharing the same surface.

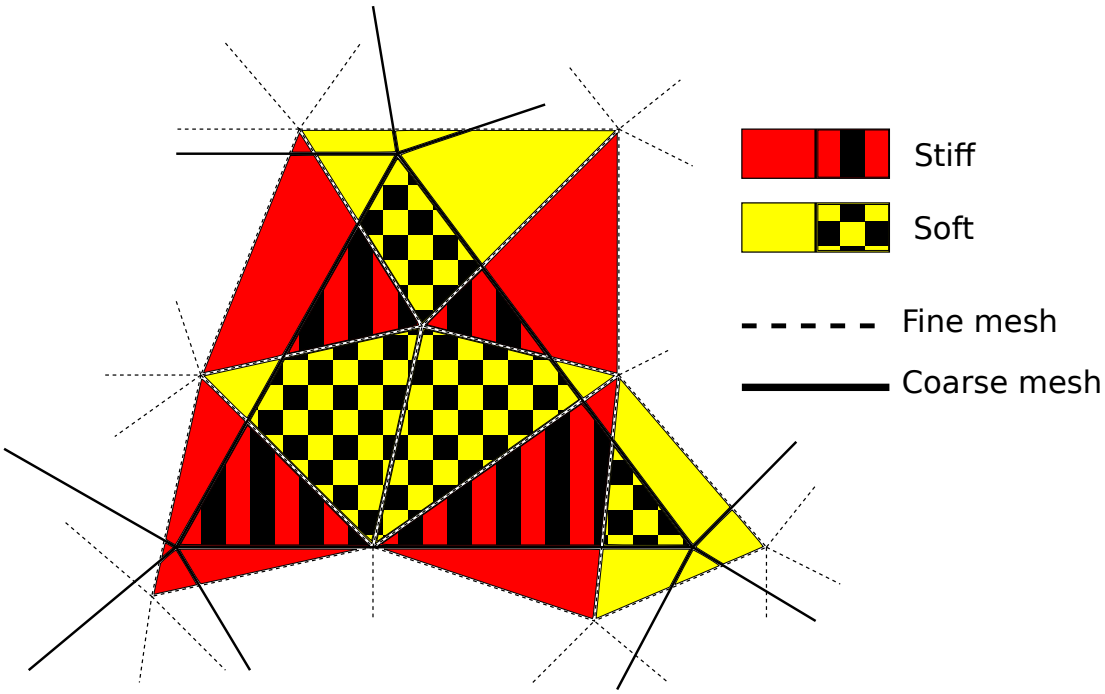


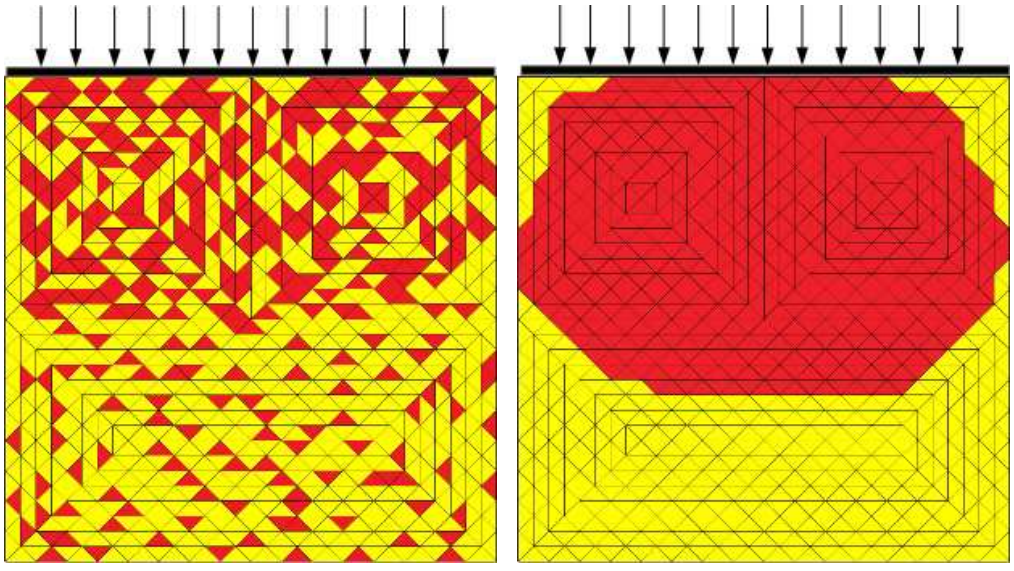
Figure 3.4 – Computation of the intersection between triangles from the finer resolution (dotted lines) and from the coarser resolution (full lines). The area between the intersection with stiff elements (in red/stripes) and soft elements (in yellow/squares) is compared in order to determine the elasticity of the coarse element.

Using this comparison matrix, the elasticity of the coarse elements is determined as the elasticity of the elements occupying the maximum space in the element. This process is illustrated in Figure 3.4. This approach has the advantage of preserving the number of elasticities from the fine aggregated object. An alternate approach computing the average elasticity value from the fine elements inside the coarse element was also considered, yet preliminary tests showed no significant improvement of this approach. Figure 3.2, Cube 5 shows the final result, with a coarser mesh. While the result is different from the original cube, an evaluation is performed using a perception-based quality criterion, as explained in Section 3. It shows in most cases a non-significant perceptual difference between the two objects.

3 Evaluation

3.1 Methodology

We assess the fact that an object coarsened by our method is sufficiently close from the original one, from a haptic perception point-of-view. The goal is to determine whether there is a difference in perception during the interaction with the object, in original and simplified state. In order to evaluate this, the initial accurate object is compared to the simplified one, for example both cubes from Figure 3.5, using forces obtained through the interaction with both objects. A perception-based criterion is used, which means that in order for the simplification to be valid, the difference between the two forces should not be noticeable. A usual metric in order to enforce this is the Just Noticeable Difference (JND), which is the maximum relative difference between two stimuli – here forces – under which there is not perceived difference. For forces, the JND is known to be close to 10% [Jones and Hunter, 1990], which means that if $\|\mathbf{f}_{\text{initial}} - \mathbf{f}_{\text{simplified}}\| \leq 10\%\|\mathbf{f}_{\text{initial}}\|$, there is no perceived difference of forces during the interaction between the original accurate object and the simplified one. We later refer to *acceptable difference*, a difference to the reference inferior to the JND, and *noticeable difference*, a difference to the reference superior to the JND. If the force from the simplified object has a difference to the force from the initial object inferior to the JND, we consider that our method has been successful.



(a) Cube before phase P1 **(b)** Cube after phase P3

Figure 3.5 – Scenario 1 close-up: the cube is compressed from a pressure on the top surface. The left cube (the input cube) has a majority of stiff (red) tetrahedra on the top part, and on the right the corresponding aggregated version, generated by phase P3.

In order to evaluate it, our method is used on several scenarios, representing general

distribution patterns, and from which most distributions can be deduced. All scenarios involve the interaction with a cube in which tetrahedra are associated with various Young Modulus. The cube is a $10cm \times 10cm \times 10cm$ cube, Poisson ratio is 0.3, and the total mass is $1kg$. Each scenario has a specific elasticity distribution pattern. For each scenario, the reaction force is recorded on the interaction with a reference fine-resolution cube, without the method being applied. The method is then applied, on various resolution levels of the mesh. Each recorded force is then compared to the reference one. The reference is also compared to the force corresponding with the interaction with the fully homogenized cube, which represents the simplest pattern from the initial configuration.

The scenarios involve a distribution of soft tetrahedra, with elasticity values $2kPa \leq E_{soft} \leq 6kPa$, and stiffer tetrahedra, with elasticity $38kPa \leq E_{stiff} \leq 42kPa$, except for the last scenario, involving a random distribution of tetrahedra with elasticities $4kPa \leq E \leq 200kPa$. All the distributions are displayed in Figure 3.6. A stiff zone contains a majority (2/3) of stiff elements, and a soft zone contains a majority (2/3) of soft elements.

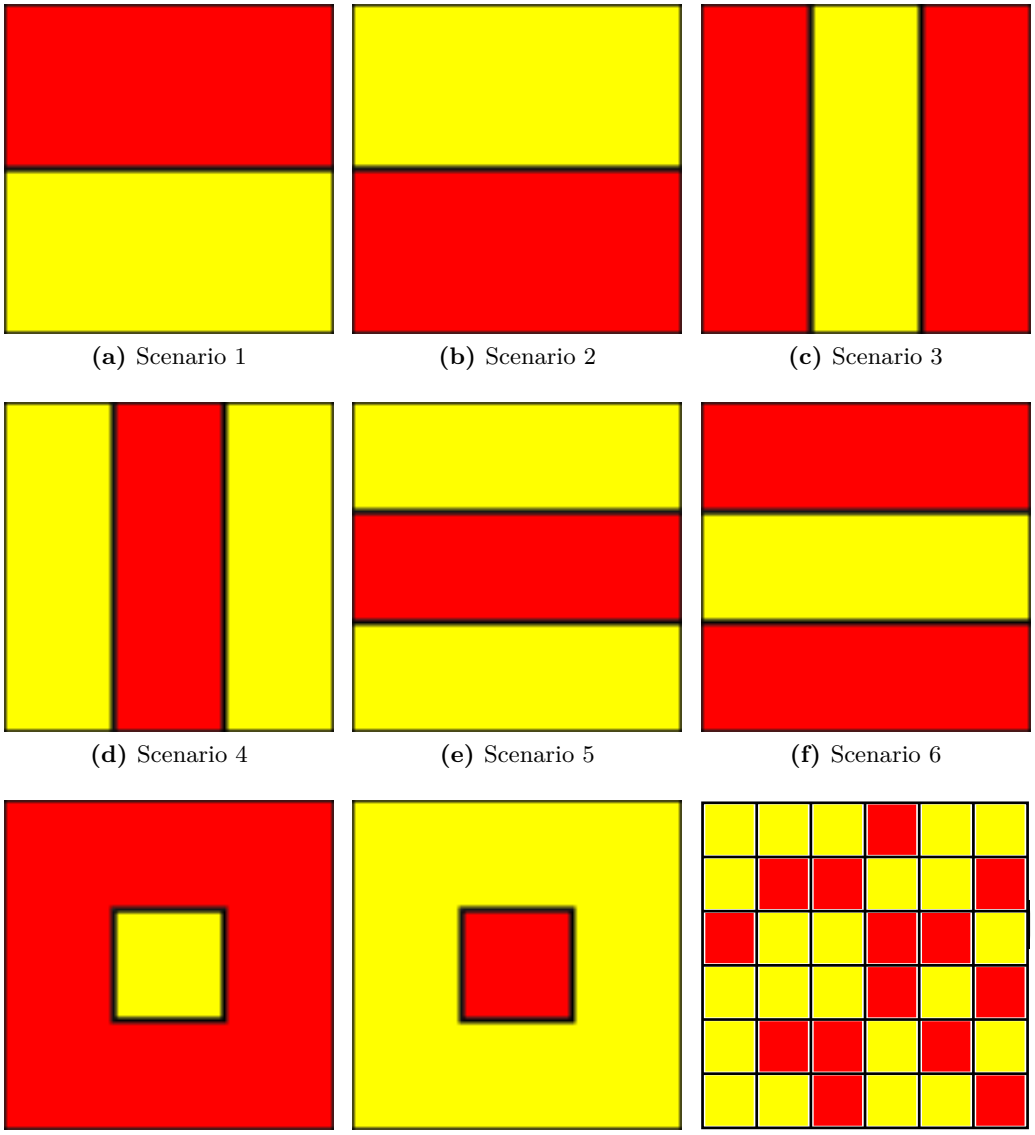
In each case, the cube is compressed from the top surface by 10% of it height over 15 seconds, followed by a static phase for 5 seconds, in order to reach equilibrium state, with corresponding force. The tested resolutions are cubes with 6,000, 3,072 and 1,296 tetrahedra. These cubes were obtained with a decomposition of the initial cube into $n \times n \times n$ smaller cubes, each divided into 6 tetrahedra. The process is illustrated in Figure 3.5, with the scenario involving a majority of stiff tetrahedra on the top part of the cube. An example of the recorded forces can be seen in Figure 3.7. The obtained behavior is really stable and consistent over the simulation, i.e. the ratio between all forces remains similar over the simulation.

As a reference, we also recorded the forces with a completely homogeneous cube, at the different mesh resolutions, in order to evaluate the level of stiffening caused by the reduction of the number of DoFs during the coarsening and to determine whether this phenomenon should explicitly be taken into account during this evaluation. Figure 3.8 shows the final force obtained with various resolutions. Each resolution correspond to a $n \times n \times n$ decomposition of the initial cube. It shows that at very coarse resolution there is a stiffening effect, but for the chosen range of resolutions the difference in force is very small, inferior to 2%, and the stiffening effect can thus be considered negligible.

Our method was implemented using SOFA framework [Allard et al., 2007], using corotational FEM for the deformation [Müller and Gross, 2004]. It was run on a PC (CPU : Intel Core i7 - 4800MQ 2.7 GHz, GPU : NVIDIA Quadro K3100M, Memory : 16GB).

3.2 Results

Performance of the simulation with the computation time required for the deformation is presented in Table 3.1. Computation time is compared between 3 mesh resolutions, the reference object has 6,000 tetrahedra, and meshes at a coarser resolution contain respectively 3,072 and 1,296 tetrahedra. The recorded interaction is a compression of the object, depicted in Figure 3.5. As expected, the simulation gets more efficient with



(a) Scenario 1 **(b)** Scenario 2 **(c)** Scenario 3

(d) Scenario 4 **(e)** Scenario 5 **(f)** Scenario 6

(g) Scenario 7 **(h)** Scenario 8 **(i)** Scenario 9

Figure 3.6 – 2D Illustration of the 9 scenarios with different elasticity distributions chosen to evaluate the method. The red part represents a stiff zone, and the yellow part represents a soft zone. Scenario 9 is a more heterogeneous distribution using a randomized filling of the cube. The presented squares are a cross-section from the actual cubes, for a force applied on the top.

a lower number of tetrahedra in the mesh. The coarsest resolution shows a speedup factor of more than 6 compared to the reference. The computation time is independent from the elasticity distribution inside the object.

The full results of the forces computed with the previously described scenarios can be found in Table 3.2. In addition to the reference force and the force recorded from the fully homogenized cube, several levels of precision of the method are presented, each in all 3 resolutions. Since the different scenarios involve two different ranges of elasticity

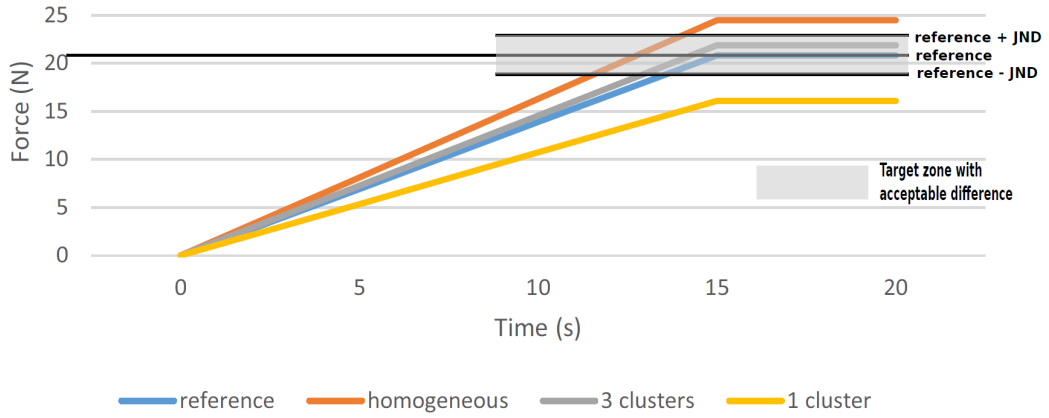


Figure 3.7 – Example of recorded forces for Scenario 2: reference force and forces with a fully homogenized cube, and using 1 and 3 clusters. The JND in force is also displayed. Only the force with 3 clusters is found here in acceptable perceptual range.

Table 3.1 – Average computation time required for the deformation of the cube with various numbers of tetrahedra. The mesh for the reference object has 6,000 tetrahedra, compared to 3,072 and 1,296 for the coarser object simplified by our method.

	reference	our method	
Number of Tetrahedra	6,000	3,072	1,296
Computation time (ms)	13.0	5.2	2.2

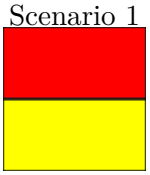
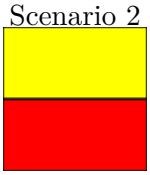
(stiff elements and soft elements), the results using the method are all computed using 2 elasticity bins. The results are compared at each resolution using 1, 3 and 6 clusters for each elasticity. The results are displayed as a percentage difference to the reference force. As a reference, forces recorded on fully homogeneous cubes for these resolutions displayed force difference inferior to 2%.

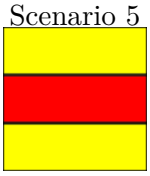
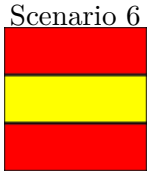
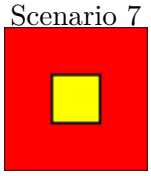
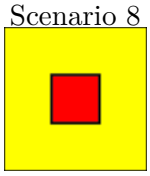
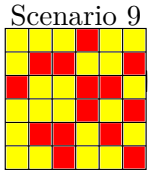
From these data, it is possible to draw several conclusions:

- Just homogenizing does not produce an acceptable force in most of the cases, as shown in Table 3.2 Line 3. In most of these cases, the homogenized cube is much stiffer than the reference cube, which justifies the use of our method;
- The force is very stable from one resolution to another, independently from the number of clusters used or the scenario involved.

More dependent on the initial distribution, a few observations can be made, representing the different general behavior of our method. In most cases, due to the simplicity of certain patterns, 3 clusters are enough to reproduce a force similar to the original one. For instance, scenarios 1, 2 and 5 have the same behavior. The force on the homogenized cube is above the reference force by an amount exceeding the JND, the force with only one cluster not important enough, and the force with 3 clusters is really close from the reference force. While the homogenized elasticity is too important, one cluster produces a large volume with a too low stiffness, creating a too low force. Those

Table 3.2 – Percentage force difference compared to the reference for each scenario, for the homogenized cube, and under the different conditions of clusters (1, 3 and 6) and resolutions (6,000, 3,072 and 1,296 tetrahedra). Green indicates an acceptable difference, and red a noticeable difference to the reference force. The threshold used to discriminate acceptable difference and noticeable difference is the JND in force, i.e. 10%.

Tetrahedra	Clusters	Scenario 1 	Scenario 2 	Scenario 3 	Scenario 4 
6,000	reference	20.7116	20.8187	25.0896	19.8527
6,000	homogenized	+19.6%	+17.6%	+9.6%	+6.5%
6,000	1	-22.7%	-22.7%	-35.1%	-8.5%
3,072	1	-22.6%	-22.3%	-35.6%	-9.4%
1,296	1	-15.9%	-16.6%	-35.1%	-8.4%
6,000	3	-2.1%	+5.1%	-16.4%	-4.6%
3,072	3	-4.4%	+5.0%	-19.0%	-5.6%
1,296	3	-0.8%	+6.7%	-14.9%	-4.4%
6,000	6	+6.6%	-0.0%	-5.8%	-5.8%
3,072	6	+6.3%	-0.6%	-7.3%	-7.3%
1,296	6	+6.8%	+0.5%	-4.4%	-4.4%

Tet.	Cl.	Scenario 5 	Scenario 6 	Scenario 7 	Scenario 8 	Scenario 9 
6,000	ref.	18.9669	22.6011	27.9015	16.5248	107.695
6,000	hom.	+16.2%	+16.7%	+6.3%	+13.0%	+3.4%
6,000	1	-30.5%	-12.1%	+2.2%	-27.9%	-5.1%
3,072	1	-32.8%	-10.0%	+2.5%	-31.4%	-5.8%
1,296	1	-30.4%	-8.3%	+3.0%	-29.4%	-5.7%
6,000	3	-4.6%	-16.9%	+0.1%	-26.7%	-3.6%
3,072	3	-8.5%	-19.0%	-1.7%	-29.8%	-4.6%
1,296	3	-4.3%	-16.7%	+0.3%	-26.2%	-3.9%
6,000	6	-5.7%	+9.5%	+3.6%	-21.2%	-3.7%
3,072	6	-6.4%	+7.6%	+2.4%	-24.0%	-5.4%
1,296	6	-3.3%	+8.5%	+2.1%	-22.7%	-4.0%

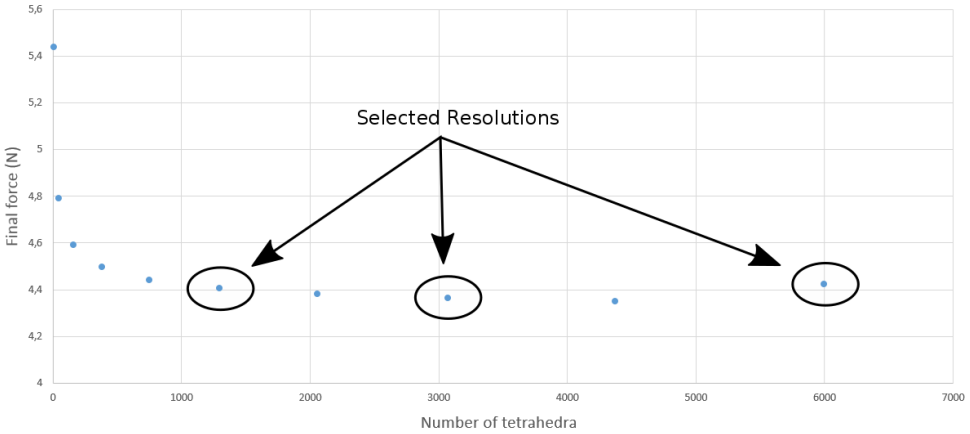


Figure 3.8 – Final force when compressing the homogeneous cube at various resolutions of the cube. The different resolutions correspond to different n values when decomposing the initial cube into $n \times n \times n$ smaller cubes. While the stiffening effect appears at really low resolution (less than 400 tetrahedra), the final force varies very little between the chosen resolutions.

scenarios have in common to have a uniform elasticity distribution on the interaction surface, and to have a large enough soft volume in the center in order to compensate for the post-aggregation soft region.

Since stiff elements are aggregated first, leaving the remaining space for soft elements, scenarios with an important proportion of stiff elements require a more important precision for the algorithm. For instance, scenarios 3 and 6 both require 6 clusters in each elasticity value to provide acceptable results. Conversely, scenarios with a great proportion of soft elements require a lesser precision of our method in order to achieve satisfactory results, such as scenarios 4 and 5.

Having a great majority of the object with a unique elasticity and a small portion of a different stiffness inside does not benefit much from our method. In the case of a mostly stiff object, such as Scenario 7, a small part of soft material inside goes completely undetected and any simplification returns an acceptable result. Conversely, a stiff element inside a mostly soft object such as Scenario 8 remains challenging for our method. While an existing stiffer core makes the homogenized cube too stiff, no reasonable (under 10) number of clusters can re-create a distribution close to the initial one, because the initial soft part included stiff elements that are later aggregated. This creates an external softer part that produces forces really inferior to the reference one. A completely random distribution (Scenario 9) does not benefit much from an important precision either, since every distribution provides a good result, because there was no clear pattern to recover, and this distribution does not create uniform soft parts that would make the homogenized cube too stiff.

As a summary, except for the case of small stiff elements inside the object, provided the number of clusters is chosen accordingly to the complexity of the initial distribution, our method provides a simplification of the elasticity that is not noticeable, yet allows an important coarsening of the object, leading to 6 times faster simulation.

4 Illustrative use case: cooking scenario

In order to illustrate our method, we implemented a use case, of a steak being cooked. The heterogeneous object, the steak, is attached to a pan, and interaction is possible through a cooking tool, a spatula, as depicted in Figure 3.9. The video illustrating the use case can be found at the following address:

http://people.irisa.fr/Benoit.Le_Gouis/Videos/VRIPHYS_2017.mp4

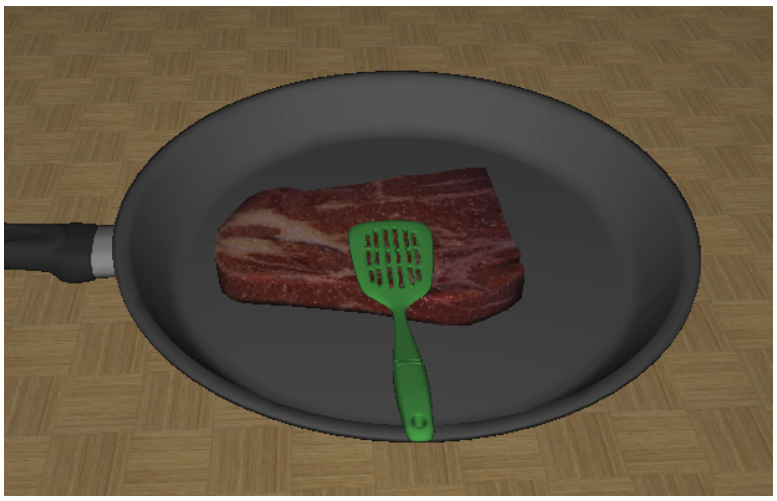


Figure 3.9 – Use case: A steak on a pan being cooked. The user controls a spatula to interact with the steak.

Haptic interaction for this use case is achieved using a Geomagic Touch (3D Systems, Rock Hill, USA), and implemented using again the SOFA framework, as depicted in Figure 3.10. We implemented an impedance haptic coupling as explained in [Lin and Otaduy, 2008]. The position of the haptic device defines a constraint on the deformable object. The force is then computed based on the deformation of the object, and returned to the user by the haptic device.

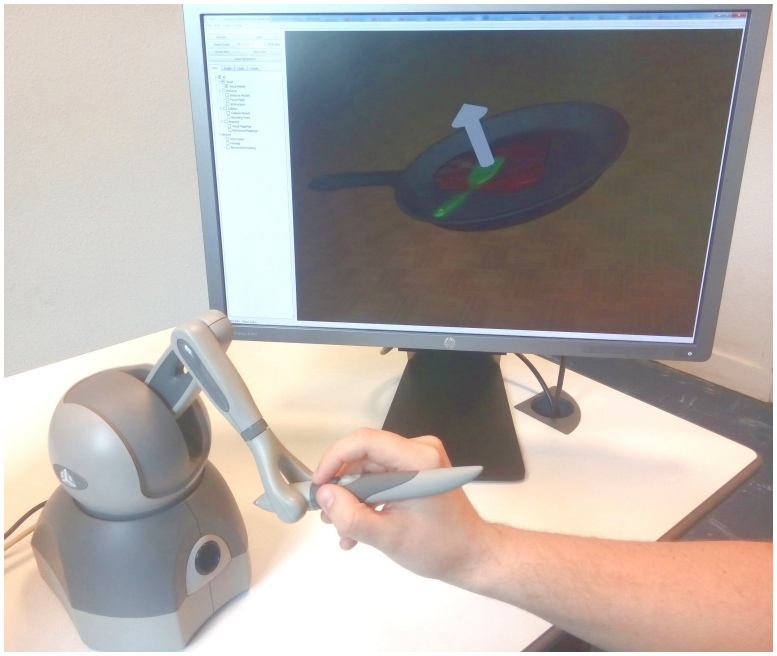
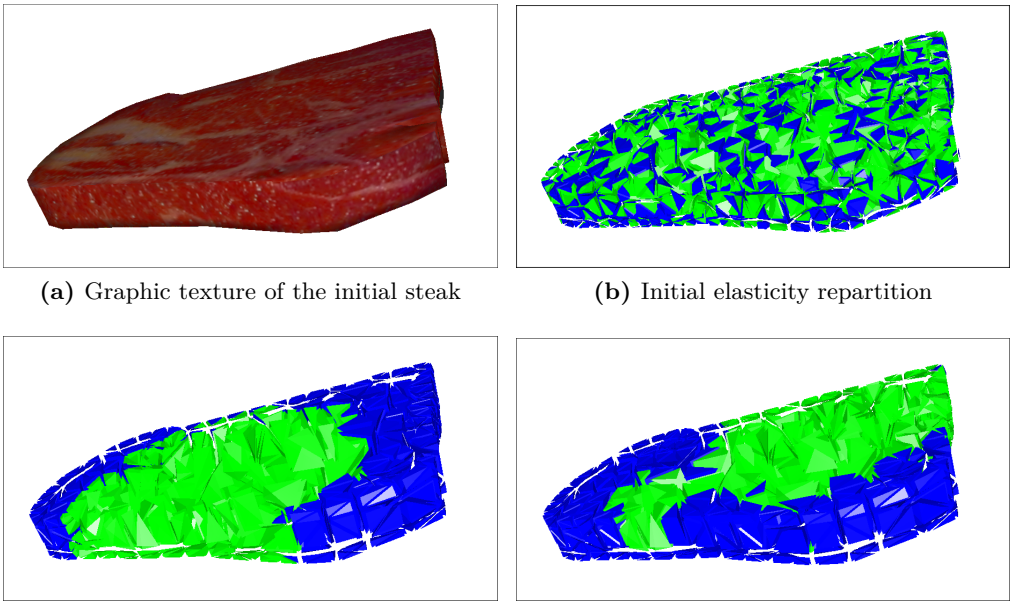


Figure 3.10 – The full setup: the user can interact with the virtual steak using a Geomagic Touch.

Two different resolutions are used for the steak, 2,894 and 1,207 tetrahedra, as depicted in Figure 3.11.



(a) Graphic texture of the initial steak

(b) Initial elasticity repartition

(c) 1 cluster

(d) 3 clusters

Figure 3.11 – Different resolutions and numbers of clusters achieved by our method: The initial textured object is shown in a) with the corresponding elasticity distribution in b). Green represents stiffer tetrahedra, and blue softer tetrahedra. The coarsened mesh using 1 cluster is shown in c), and using 3 clusters in d).

Two levels of precision of our methods are compared, using 1 and 3 clusters for each elasticity. A cooking steak is expected to have a greater concentration of stiff elements on the bottom part, corresponding to Scenario 2. The interaction consists in a progressive constraint on the top of the steak by the spatula, similar to the interaction used in the previous section. Results show an improvement in performances, from 70Hz on the fine mesh to 120Hz on the coarse mesh. Interaction with the object with 3 clusters creates an acceptable difference (5% difference), while there was a noticeable difference in the interaction with the object with only 1 cluster (50% difference), as depicted in Figure 3.12.

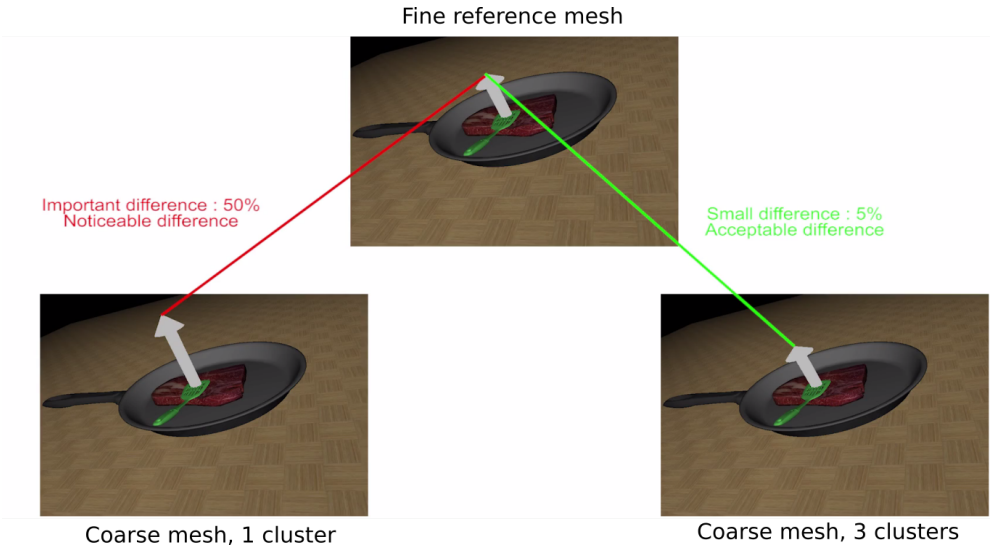


Figure 3.12 – End force comparison between the fine resolution object and the two coarse objects, using our method with respectively 1 and 3 elasticity clusters. At the coarser resolution, the object with 1 cluster shows a noticeable difference (50%), whereas the object with 3 elasticity clusters shows an acceptable difference (5%).

5 Conclusion

In this chapter, we presented a novel multiresolution approach for haptic interaction with heterogeneous deformable objects. Our method addresses the issue of elasticity attribution for the geometric coarsening of heterogeneous objects. It is based on the design of elasticity clusters that create large homogeneous volumes inside the object, making the attribution more straightforward. While our method enables to improve computation time performances, it still keeps important features of the elasticity distribution of the original object. We evaluated our method on diverse scenarios using a perception-based quality criterion. We found that the recorded force is independent from the multiresolution level, thus allowing the use of a much coarser geometry, and subsequent important gain in performances, at a non-noticeable perceptual cost.

As of today, this method suffer a few limitations. For instance, it works best with large interaction surfaces, since a too localized interaction would be heavily influenced by a redistribution of the elasticity. The size of the tool used to interact must thus be

of comparable size of the object –or the part of the object – that has been coarsened using our method. This also means that interaction involving topology changes such as cutting, by nature quite local to the location of the interaction, are not compatible with such simplification of the object.

As a conclusion, this contribution shows a method that can improve computation performances for haptic interaction with heterogeneous objects and relies on a multi-resolution approach, and this with little haptic perceptual cost. It thus addresses the computation complexity issue that arises when designing haptic interaction with virtual environments, in the case of heterogeneous deformable objects.

Haptic Tearing of Deformable Surfaces

4

Contents

1 Introduction

Thin deformable surfaces are very common in everyday life, from cloth to paper, bags and food. When manipulated, these deformable objects often undergo topology modifications through tearing phenomena. The real-time simulation of such phenomena in virtual environments has therefore many applications, from surgery training to video games and animation. It remains however very computationally-demanding, preventing their use for many interactive applications.

In that context, after the model-oriented contribution presented in the previous chapter for the interaction with heterogeneous objects, this contribution addresses issues related to another origin of complexity for deformable objects, topology changes. While there has been a great focus on cutting simulation involving haptic interaction, surface tearing has received much less attention. For a tearing simulation to achieve sufficient performances to allow for haptic interaction, several components are required,

representing the main bottlenecks in terms of computation cost, namely collision detection involving surfacic deformable objects, and the tearing simulation itself. This simulation-oriented contribution thus incorporates these elements to build an efficient haptic tearing simulation for haptic interaction.

In this chapter, we propose a fully-functional interaction pipeline for physically-based simulation of tearing with haptic rendering. It relies on the first hand on a novel and efficient collision detection method for thin deformable objects, including auto-collisions and collision with the environment, and on the other hand on a tearing simulation based on an efficient Finite Element Method (FEM) compatible with haptic rates. We also provide a haptic interaction scheme allowing the use of multiple haptic devices, such as for two-handed haptic tearing simulation.

This work has been done in cooperation with François Lehericey for the collision detection part. Please refer to [Lehericey, 2016] for in-depth explanation of the collision detection method used for this contribution.

The remainder of this chapter is organized as follows. The general pipeline for haptic tearing simulation is presented in section 2. Section 3 details the collision detection method used in our pipeline. The tearing method is then described in section 4, and the haptic coupling is later presented in section 5. Several use-cases involving haptic tearing are proposed in section 6, as well as their corresponding computation time performances.

2 General description of our haptic tearing approach

In our approach, the user can interact in real-time with a virtual environment composed of thin deformable objects and receive multi-sensory feedback through the use of screens and several haptic interfaces. Our physically-based simulator is composed of two main components:

1. a novel collision detection method to provide information of contact between the deformable surface and its environment, as well as auto-collisions;

2. FEM simulation including a tearing model for thin deformable objects.

During the simulation, collision response dictates the behavior of the surface in contact with its environment. Based on this contact information and on the user input, deformation of the surface is determined by the physically-based model, that can handle tearing phenomena. Feedback is then provided to the user using the information from the physically-based simulation. In our approach, a haptic coupling is implemented to provide a bimanual haptic feedback, depicted in Figure 4.1.

In order to achieve interactive simulation rate, each of the steps of the simulation pipeline must be performed with minimal computational cost. Collision detection and physically-based rendering of tearing are the main bottlenecks in terms of computation time, and are therefore hereafter described.

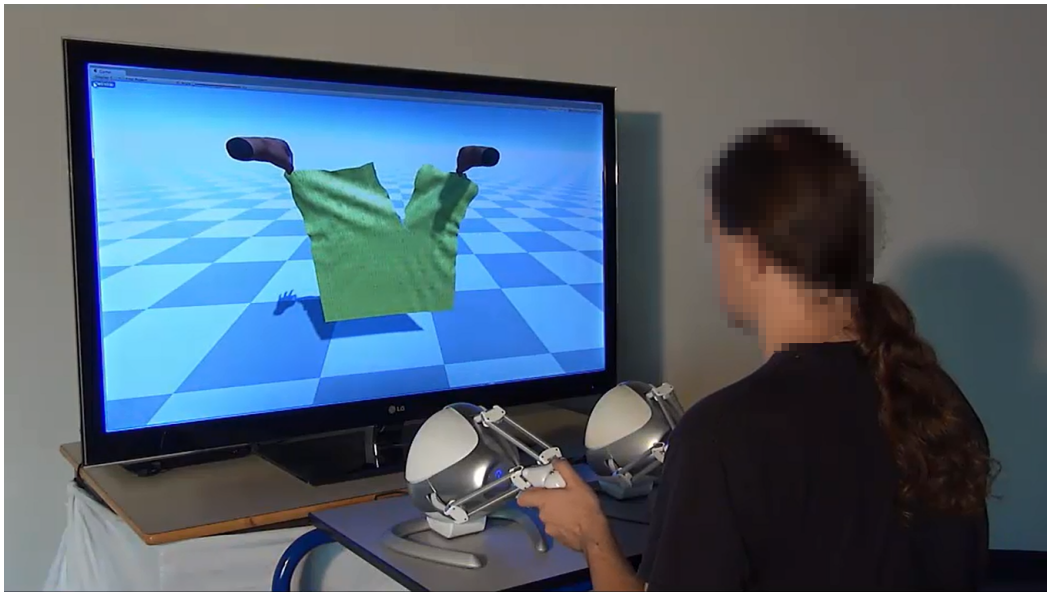


Figure 4.1 – The setup for our bimanual haptic tearing method: a user can tear a deformable surface using two haptic devices.

3 Collision detection for surface meshes using a novel clustering formulation

3.1 Related work on collision detection

Besides the tearing simulation itself, the simulation pipeline for tearing phenomena should handle efficient collision detection of the torn components of the virtual environments. Collision detection remains one of the main bottlenecks of physics simulation, moreover in scenes containing highly dynamic objects such as deformable objects and topological changes where the cost tends to highly increase. In the context of haptic simulations where performance is more critical than others applications, this has led to specific methods. For instance, Gregory et al. [Gregory et al., 2005] proposed a framework that uses spatial decomposition, bounding volumes hierarchy and temporal coherence to achieve collision detection on rigid objects at a frame-rate compatible with haptic simulations. In the case of deformable objects, Barbič [Barbič and James, 2008] proposed to perform collision detection by representing objects with a point-based representation and signed distance fields. To comply with the high frame-rate requirement of haptic rendering the authors proposed to use a graceful degradation of contacts: if there is not enough computation time to fully perform collision detection, the algorithm returns a reasonable answer instead of the complete one.

As explained in section 2.2.2, collision detection computes for each pair of objects which ones are colliding. This process represents a major bottleneck in terms of required computation resources, and thus requires efficient yet accurate methods, especially when haptics is involved. Collision detection with deformable objects add an additional challenge, since the collision model of the object cannot be deduced only from the

position and orientation of the object, and must be updated to take the deformation of the object into account. This represents a major computational challenge, especially during the middle phase of the collision detection process.

In this contribution, the proposed method is to optimize the collision detection by regrouping the vertices in clusters [Lehericey, 2016]. Using a measurement of the relative displacements between the clusters, we can identify the vertices where a full collision detection needs to be performed. Based on this method, we are thus able to propose a full-functional interaction pipeline for simulating tearing phenomena on deformable surfaces with haptic rendering.

3.2 Method overview

In order to simulate a thin deformable surface, collision detection queries are required at each time step of the simulation. We chose to use an image-based method to perform collision detection since these methods are more adapted to highly dynamic objects such as a piece of cloth that can undergo deformations and topological changes. In particular, we used a ray-tracing based method proposed by [Lehericey et al., 2015] since it is able to handle both surfacic objects (in the present case for the deformable surface) and volumetric objects (for the surrounding environment in our case).

For haptic rendering, the required frame-rate of the physically-based simulation is higher than for visual feedback only. To complete collision detection within efficient frame-rates, temporal coherency can first be used to accelerate the collision queries. Temporal coherency is a property that comes from the continuity of the motion of the objects in the simulation. It states that the positions and geometric configurations in space of the objects will be similar between successive time steps. To exploit this property, an incremental method is more profitable to the high frame-rates. Lehericey et al. [Lehericey et al., 2013] proposed such an approach for rigid objects, in order to reuse the results of the collision detection from the last step and then update them incrementally instead of performing collision detection from scratch at each time-step.

For this contribution, we propose to extend this approach to deformable objects, such as thin deformable material. To decide if an incremental method can be used, we measure temporal coherency between objects by measuring the quantity of relative displacement between pairs of objects. If objects have low relative displacement over time, e.g. the relative displacement stays under a threshold called *displacementThreshold*, then an incremental method is used to compute collision detection, otherwise a non-incremental method is used. The quantity of relative displacement is measured by studying the evolution of the frame of reference between pairs of objects. However, it does not take into account the internal deformation of the objects. We thus propose a novel method to enable us to measure the quantity of relative displacement between two deformable objects (or between a rigid and a deformable objects), that takes into account the internal deformation of the objects. This new method does not perform a global measurement of relative displacements between the two objects but works on a decomposition of the objects called clusters. It is thus able to detect temporal coherency locally even if the two whole objects do not exhibit strong temporal coherency. In the following paragraphs, we detail the different steps of our approach.

3.3 Decomposition of objects in clusters

The first step of our method consists in decomposing the deformable objects in clusters. The cluster decomposition is motivated by two computation-intensive and memory-consuming issues: (1) contrary to rigid objects, the displacement of each individual vertex of a deformable object is independent and we therefore need to measure each displacement, (2) the history of the position of each vertex over time needs to be stored, in order to know when the relative displacement exceeds the threshold *displacementThreshold*. To reduce the time and memory complexity related to displacement measurement, we propose to measure the displacement of clusters of vertices instead of measuring the displacement of each vertex individually. The use of clusters is motivated by the observation that neighbouring vertices tend to have similar movements because of spatial coherency.

To decompose the object into clusters of vertices, we use the k-means clustering method. The number of vertices per cluster is empirically fixed in our case (100 vertices per cluster) as we have homogeneous deformable objects. We could however adapt it depending on the object properties. As a distance metric for the k-means, we do not use the Euclidean distance because it does not guarantee that the vertices will still be close in space if a deformation occurs. Instead, we use a metric based on the distance on the surface of the object which stays valid even after the objects have undergone some deformation. Our distance metric between two vertices is the length of the shortest path between the two vertices through the edges. Furthermore, this metric guarantees that no cluster will be disjointed. Figure 4.2 gives an example of the decomposition into clusters of cloth.

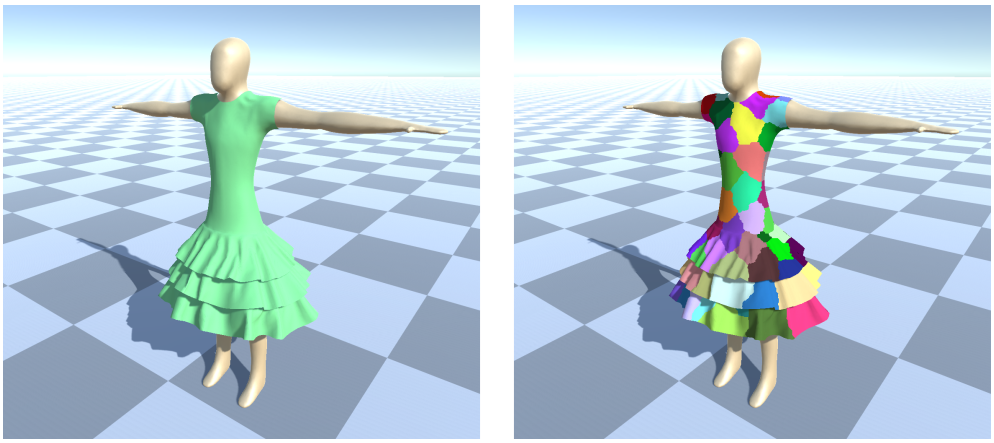


Figure 4.2 – Example of the decomposition of a dress (left) into clusters (right). The dress is composed of 11,000 vertices and is decomposed into 110 clusters.

3.4 Relative displacement measurement of clustered objects

The second step of our method is to compute the relative displacement measurement between clustered objects. The measurement is itself performed in three steps:

- Measurement of absolute displacement of clusters.

- Detection of neighbouring clusters.
- Measurement of relative displacement of neighboring clusters.

The first two steps are independent. The goal of the first step is to reduce the complexity of the movement of all the vertices in the cluster into a more easily manageable form. The second step is used to decide on which pairs of clusters we need to measure the relative displacement from (to avoid performing unnecessary measure on pairs that are far away). The third step combines the output of the two first steps to compute the relative displacement on the selected pairs of clusters.

3.4.1 Measurement of absolute displacement of clusters

To obtain a general information about the displacement within a cluster, we first determine the absolute displacement of each cluster. We measure the relative displacement only from the previous time-step and we do not need to store the values of the previous steps.

To reduce the complexity of measurement, we propose to measure the average displacement of the vertices and the standard deviation from the average displacement. For each cluster c, the average displacement $dis_{avg}(c) \in \mathbf{R}^3$ is a vector containing the main direction and the amplitude of the displacement, the standard deviation $dis_{dev}(c) \in \mathbf{R}$ is a scalar measuring the rotation and deformation of the cluster.

3.4.2 Detection of neighbouring clusters

In this step, we list the pairs of neighbouring clusters. This is done to perform culling on the next steps as we only need to perform a measurement of relative displacement and collision detection on very close clusters.

To list the pairs of close clusters, we propose to use a broad-phase collision detection algorithm. We propose to compute for each cluster an Axis-Aligned Bounding Box (AABB) and a pair of clusters is considered close if their AABBs are in collision. Figure 4.3 illustrates the detection of close clusters in the case of auto-collision.

When a new pair of clusters is detected (i.e. a pair that was not detected in the previous time-step), their relative displacement is initialized to zero. A full collision detection method is used when their relative displacement reaches the displacement threshold *displacementThreshold*. As two clusters should not enter in collision before a full collision detection algorithm is used on them, we extend the AABB of each cluster by a distance of *displacementThreshold*/2 in every direction to enforce this property.

3.4.3 Measurement of relative displacement of neighboring clusters

The measure of relative displacement is performed on each pair of close clusters. For each pair of clusters (c_1;c_2), the relative displacement in the current time-step t is computed from the absolute displacement of each cluster:

$$CurrRelDis_{avg}(c_1, c_2)_t = dis_{avg}(c_1) - dis_{avg}(c_2) \quad (4.1)$$

$$CurrRelDis_{dev}(c_1, c_2)_t = dis_{dev}(c_1) + dis_{dev}(c_2) \quad (4.2)$$

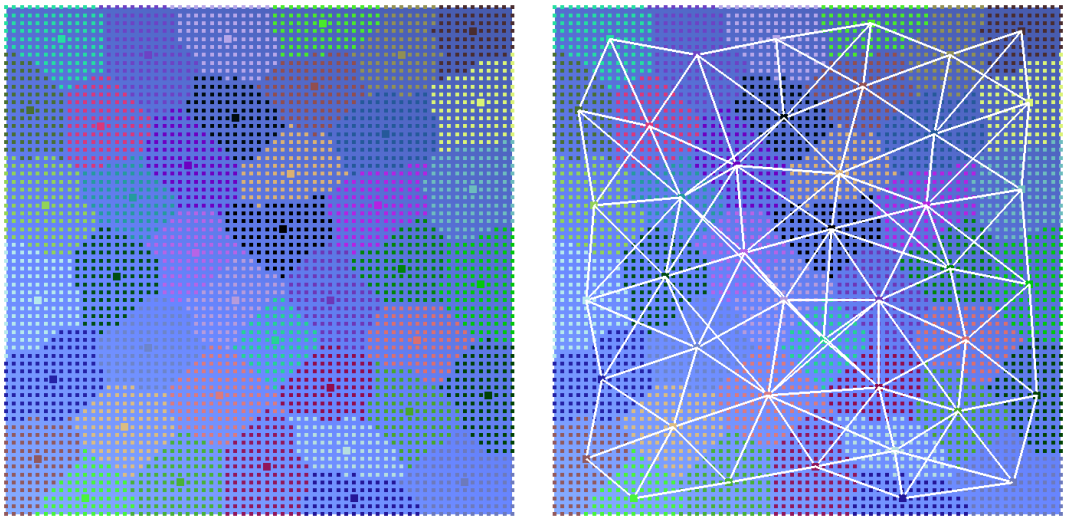


Figure 4.3 – Example of detection of close clusters. On the left, a squared piece of cloth (4,225 vertices) is decomposed into 42 clusters (the vertices are colored according to their affiliation to a cluster). On the right, the relation of closeness between pairs of clusters is displayed with white lines (which indicates collision between the cluster's AABBs in this specific geometric configuration). In this example, we have 121 pairs of close clusters.

This measure of relative displacement for the current time-step ($CurrRelDis$) is then added to the complete measure of relative displacement ($RelDis$) which stores the accumulated amount of relative displacement since the last use of a full collision detection algorithm (the formula is the same for both the average and the standard deviation):

$$RelDis(c_1, c_2)_t = RelDis(c_1, c_2)_{t-1} + CurrRelDis(c_1, c_2)_t \tag{4.3}$$

The amplitude of the relative displacement $\|RelDis(c_1, c_2)\|$ can then be computed:

$$\|RelDis(c_1, c_2)\| = \|RelDis_{avg}(c_1, c_2)\| + RelDis_{dev}(c_1, c_2) \tag{4.4}$$

When this amplitude exceeds $displacementThreshold$ a full collision detection algorithm is used and $RelDis(c_1, c_2)$ is reset to zero, otherwise an incremental collision detection algorithm is used. This distance has been empirically chosen as the best among several implemented metrics.

Our method for collision detection allows to measure more precisely the relative displacement between deformable objects, thus easing the use of incremental methods for collision detection. In particular our method can detect that two objects with similar velocities have low relative displacement, whereas an absolute displacement measurement would indicate that the two objects are undergoing high displacements.

In order to assert the efficiency of our method, we used a moving character wearing clothes (see Figure 5). If the character is immobile, a full collision detection takes on average 3.65ms (1.22ms standard deviation SD), with an absolute displacement measure, and 3.37ms ($SD = 0.75ms$) with a relative displacement measure, a rather similar amount. However, if the character is moving, the absolute displacement measure

goes up to 11.76ms ($SD = 2.01ms$). As expected, our relative displacement measure brings that cost back to 3.18ms ($SD = 0.46ms$), for an almost 4 times speedup, because the relative movement between the body and the t-shirt is rather small, and allows a higher usage of incremental methods in the collision detection.

4 Physically-based simulation of tearing

4.1 Related work on tearing Simulation

Tearing simulation has been first introduced in computer graphics by Terzopoulos [Terzopoulos and Fleischer, 1988]: the tear creation and propagation was driven by the level of stress and strain of the objects. This basic idea of the approach has been further used for other fracture and tearing phenomena. Tearing methods rely on the chosen cloth simulation method. There are two main approaches to the simulation of cloth, a sheet-based approach, considering the cloth as a surface, and yarn-based approaches, that represent the cloth as a weaving of numerous yarns.

4.1.1 Yarn-based approach

In this approach, cloth is represented as a set of woven yarn. The dynamics of the cloth is determined by the chosen model of yarn, as well as the collision representation between the yarn. First approaches compute the contact between the yarns, using penalty and friction forces in order to compute the response ([Kaldor et al., 2008], [Kaldor et al., 2010]). Yarns are considered as inextensible rods in order to keep the computation complexity as reasonable as possible. In order to achieve better performances, [Cirio et al., 2014] consider that yarn are always in contact, removing the need to compute collision between them. The cloth is considered as an array of contact points, that each require 5 coordinates, namely the position in space, and a sliding coefficient on each weaving direction. This enables to reduce greatly the computation cost of simulation. This method was later extended in [Cirio et al., 2017] for more complex knitted patterns. With this representation, cloth exhibit non-linear behavior close to what can be observed on real cloth, such as shear friction, or curling at the edges. As stated in [Cirio et al., 2014], tearing occurs naturally when the stress on a yarn exceeds a certain threshold.

While yarn-based produce really accurate cloth behavior, this simulation method is far from being interactive. For instance, with the more efficient method presented in [Cirio et al., 2014], some models require more than one hour of computation per second of simulation, which makes this approach incompatible with haptic interaction.

4.1.2 Sheet-based approach

In order to simulate cloth in an interactive manner, other approaches consider cloth as a deformable surface, cutting down the complexity of simulation : sheet-based approaches. As for volumetric deformable objects, mass-spring systems and FEM approaches have been proposed.

Mass-spring systems are widely used in cloth simulation, since they offer good performances, and the accuracy is often considered to be sufficient for macro-scale behavior of cloth [Baraff and Witkin, 1998]. Tearing on a cloth simulated with a mass-spring system is also quite straightforward, cutting spring that undergo a too large stress [Terzopoulos and Fleischer, 1988]. In order to provide with a more accurate simulation Metaaphanon et al. [Metaaphanon et al., 2009] aims at modeling frayed edges of woven fabrics being torn. A simple mass-spring system, the base cloth model, represents undamaged regions, and a yarn-level model where each mass particles are split in two loosely coupled particles linked to different threads of springs models the two layers of interwoven yarn of the cloth. While providing high-quality visual rendering, the several seconds required to simulate each timestep also make it unsuited to interactive applications.

FEM methods to simulate cloth tearing have also been proposed. While they all rely on a mesh separation at high stress location, these methods greatly differ by the way they interact with the object's mesh. For instance, the method of Souza et al. [Souza et al., 2014] consists in splitting one vertex of the overstretched edges to create a tear that follows the existing edges. In order to avoid creating too many visual artifacts, this method requires a fine quality mesh that can impair performances, and is thus either not visually plausible, or unsuited for haptic interaction. Pfaff et al. [Pfaff et al., 2014] based on the prior work on adaptive meshes by Narain [Narain et al., 2012] proposed to refine the mesh where a crack is likely to propagate or be created and coarsen it elsewhere. The direction of a potential crack is computed by finding the potential splitting plane for which the most stress energy would be released. While more accurate than the previous methods, this method is computationally heavier, and does not seem suitable for haptic interaction. Gingold et al. [Gingold et al., 2004] based their fracture handling on the strain tensor of each triangular face of the mesh: a face is split into two if the principal strain exceeds a material-specific threshold. The fracture splits the strained face along the direction orthogonal to the principal strain direction. This method has been extended by Allard et al. [Allard et al., 2009] for anisotropic elements, simulating fibers. This anisotropy is taken into account for the tearing propagation. Splitting triangles on the tearing path can lead to an important increase in the number of ill-shaped triangles, for example in [O'Brien and Hodgins, 1999] and [O'brien et al., 2002]. Some methods however exist to cut a mesh with a minimal amount of created triangles, and keeping the best possible quality for their shape. A review on these methods has been performed in [Wu et al., 2014]. While both [Gingold et al., 2004] and [Allard et al., 2009] provide a simple – yet accurate – model for physically-based tearing simulation, neither method discusses haptics.

4.2 Our method for efficient physically-based simulation of surface tearing

In this work, in order to get a more accurate behavior on the surface simulation, we use a sheet-based representation of the surface using FEM [Nealen et al., 2006]. As discussed in the related work, yarn-based methods provide a very accurate cloth simulation, but with about 1 hour computation per second of simulation the performances do not allow for interaction, let alone haptic interaction [Cirio et al., 2017]. Mass-spring

methods would also provide for efficient simulation, but at the expense of accuracy of the surface behavior. Our method thus relies on a co-rotational FEM for surface objects. For performance purposes, only the elastic deformation is taken into account, and no plasticity is computed. This also allowed to build the entire setup, from which improvement could be performed. FEM-based methods compute stress based on the strain of elements. In order to simulate tearing, our method is inspired by the tearing simulation performed in [Allard et al., 2009] for isotropic surfaces. Tearing occurs when the stress reaches a material-dependent threshold σ_E, as depicted in Figure 4.4. In the case of isotropic objects, the maximal stress is computed using an eigenvalue decomposition of the stress tensor. The maximal eigenvalue provides the maximal stress to be compared to the threshold. Several points can reach the threshold simultaneously, due to discrete time integration, in which case only the highest value is kept as candidate to start tearing. Once the threshold is reached, tearing starts orthogonal to the direction associated to the maximal stress, given by the eigenvector associated to the maximal eigenvalue. The orthogonal direction is chosen inside the plane around the tearing tip, and then projected on the mesh. Previous tear direction is stored in order to avoid tearing going backwards, ensuring that a tear can only propagate in the same overall direction as the previous timestep.

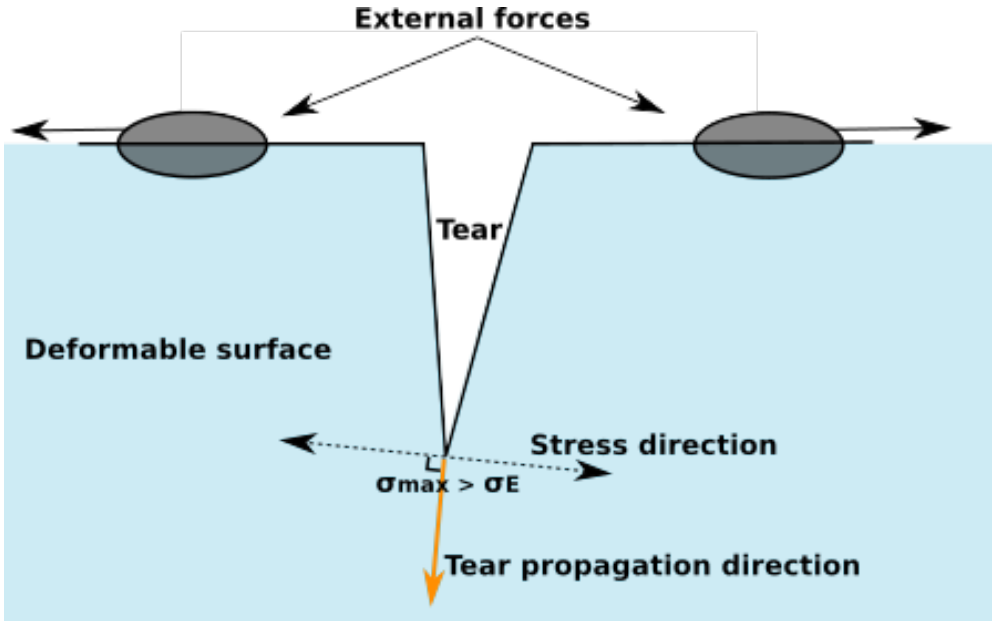


Figure 4.4 – Illustration of the tearing process: the surface is under stress generated by an external action (here two parts of the surface are pulled in opposite directions). Tearing starts when the local stress reaches the threshold σ_E, and the tear direction is orthogonal to the stress direction.

Computing the eigenvalues at each vertex of the mesh can prove costly in terms of computation. In order to address this, once started, we chose to propagate a tear only from the previous tearing tip. Stress eigenvalue decomposition is thus not performed on other vertices.

A loss of performance can also occur if too many elements are created during the tearing process. Several methods exist to address this issue. A first method would be to

separate the mesh along existing edges, thus creating no new elements, similar to what is performed in [Müller et al., 2001], but it creates visual artifacts, it is affected by the mesh resolution, and it does not provide an accurate tearing path. In our approach, a snapping method is used, such as described in [Wu et al., 2014], which locally adapts the mesh to make it match the tearing path as best as possible. As described in Figure 4.5, the snapping process computes the proximity of the mesh vertices to the tearing path, and displaces the position of the vertex if it is close enough from the tearing path, in order to reduce the number of created triangles. Furthermore, the created triangles have a much better shape, ie they have a comparable size to the existing triangles in the mesh and they are not really thin, which makes for better in the simulation. It also greatly reduce the number of created triangles – in our example, after the cut, the number of triangles is increased by 2 instead of 8, with triangles having a better shape – thus allowing for rather stable performances during the simulation.

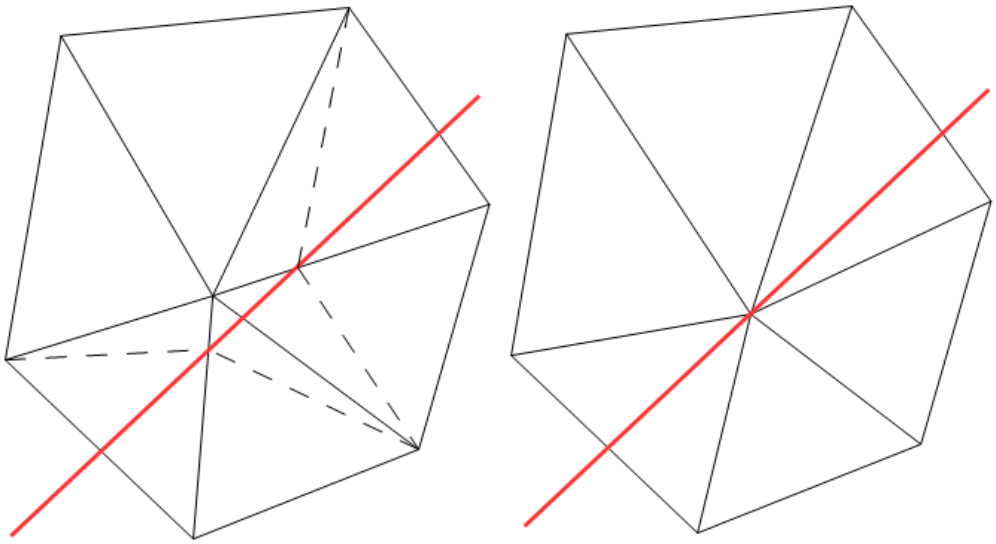


Figure 4.5 – The mesh is cut without (left) or with (right) the snapping method. The snapping method adapts the mesh to match the best the tearing path, thus creating fewer triangles, of better shape.

All these elements are gathered together to provide a tearing simulation, with performances compatible with haptic simulation. In particular, due to the snapping method, the performances are not significantly affected by the creation of new elements during the tearing process.

5 Haptic rendering

On top of our physically-based simulator, we propose a coupling scheme to provide bimanual haptic interaction to the user. In our approach, haptics is handled in a separate thread with appropriate frequency. To each haptic device is associated a constrained part of the deformable object, representing the grasped handle of the object, and a virtual proxy to this handle. The proxy position is directly coupled to the device position, at haptic rate h_{rate}, as an affine function of the device position. The handle

position is updated to the proxy position at simulation rate s_{rate}.

Force feedback is composed of several elements, given in equation (4.5). First, the force contribution of all the vertices in the handle, internal and external forces, are aggregated into a tearing force $\mathbf{F}_{tear}$, as shown in equation (4.6), updated at simulation rate. In order to provide a force at haptic rates, a proxy spring force is added, based on the relative positions of the proxy and the handle, and a damping factor is added on the velocity for stability purposes. This defines an impedance haptic scheme, such as described in [Lin and Otaduy, 2008].

$$\overbrace{\mathbf{F}_{tot}}^{h_{rate}} = \overbrace{\mathbf{F}_{tear}}^{s_{rate}} + \overbrace{\underbrace{k_{position} \times (\mathbf{x}_{proxy} - \mathbf{x}_{tool})}_{\text{Spring to the proxy position}} - \underbrace{k_{velocity} \times \mathbf{v}}_{\text{Damping}}}^{h_{rate}} \tag{4.5}$$

$$\mathbf{F}_{tear} = \sum_{i \text{ constrained}} (\mathbf{F}_{elastic}(\mathbf{x}_i) + \mathbf{F}_{external}(\mathbf{x}_i)) \tag{4.6}$$

This force is then scaled and provided as direct feedback. Such a scheme allows for a precise control of the deformation, and provides a direct force feedback. A tearing task usually involving two hands, our method allows for the use of multiple haptic devices simultaneously, in order to fully feel the forces involved in the tearing process.

6 Use-cases and performance

6.1 Implementation setup

Our method has been implemented using SOFA framework [Allard et al., 2007] and runs on a PC (CPU: Intel Xeon E5-2620 2.1 GHz, GPU: AMD FirePro W9100, Memory: 32GB). Visualization is performed with Unity (www.unity3d.com). The haptic feedback is provided through a bimanual setup with two Novint Falcons, handled with chai3d framework (www.chai3d.org), that offers a device-transparent interface for haptic devices.

6.2 Illustrative use-cases

Our approach is illustrated through two use-cases using commonly torn material, paper and cloth. Both of our illustrative use-cases use a bimanual haptic setup.

- The first use-case illustrates a paper tearing scenario and involves a banknote being torn apart. Tearing starts almost immediately, since paper has a high Young Modulus and a relatively smaller stress threshold. This scenario involves only the autocollision computation.
- The second use-case features a superman-inspired t-shirt tearing scenario. The cloth tearing involves the entire simulation pipeline, namely collision detection with the environment, cloth simulation and tearing and bimanual haptic rendering, as well as a moving body. The full contact between the cloth and the body underneath is a really challenging scenario because it involves complex shapes and thus illustrates the efficiency of the collision detection algorithm.

The use cases are depicted in Figure 4.6 and are also further illustrated in the following video:
http://people.irisa.fr/Benoit.Le_Gouis/Videos/WHC_2017.mp4

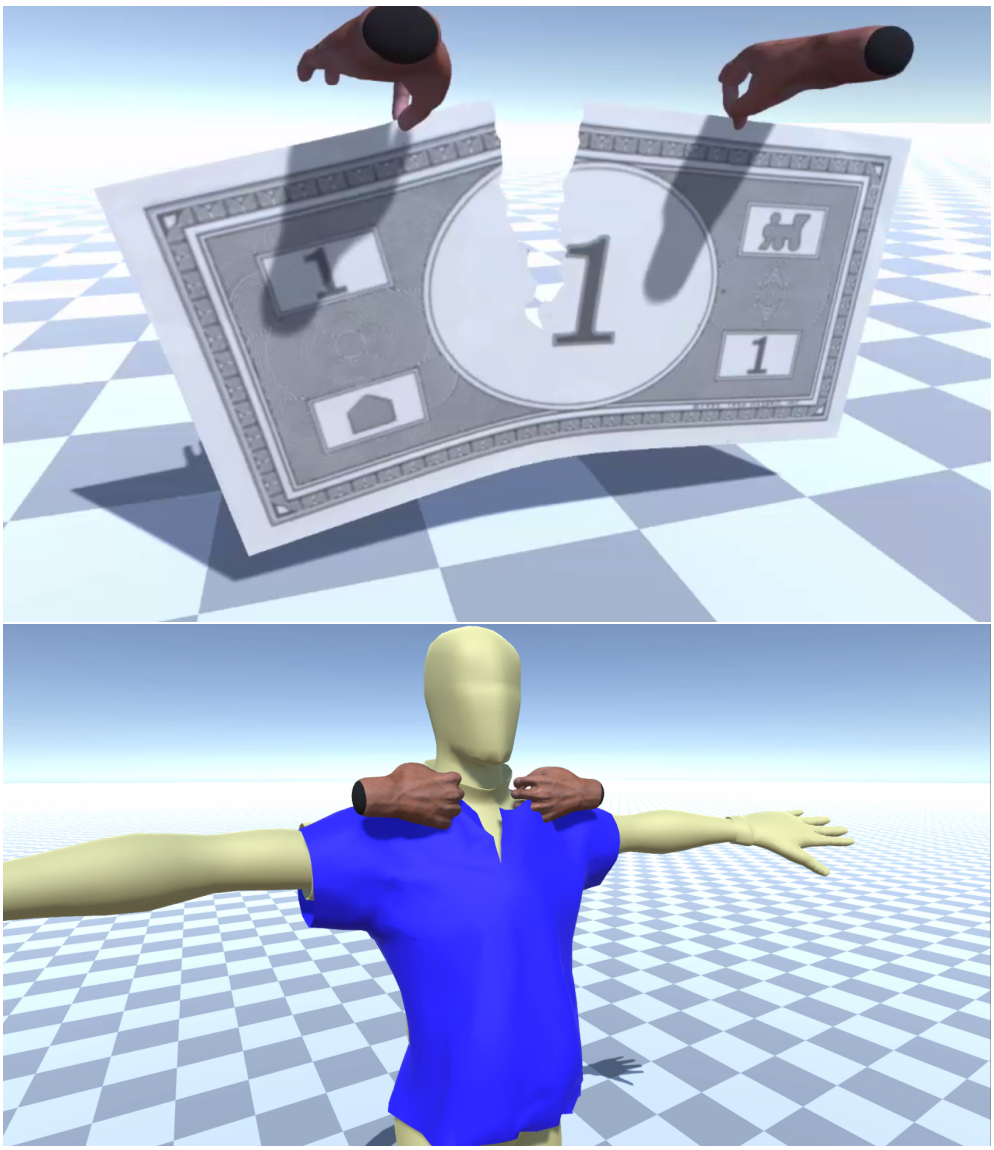


Figure 4.6 – Illustrative examples: (Top) a paper banknote is torn. (Bottom) a worn t-shirt where collision is activated for the full body is torn.

6.3 Performance

Table 4.1 shows the computation performance of our method, on several mesh sizes, corresponding to the square piece of cloth used as illustration of our setup in Figure 4.1, and the two use-cases displayed in Figure 4.6, the banknote and the t-shirt. Computation time is decomposed into three categories: collision detection, FEM deformation and tearing computation. Two time values are provided for the tearing

contribution, since there is a difference between the computation before the tearing starts, with the computation of the principal stress for each vertex, and after the mesh starts tearing, with the sole computation of the tearing path. The objective total computation time corresponds to one time-step for the chosen illustrative use-cases. The performance table shows that the choice of propagating from an initial tear only induces a significant improvement in the performances as soon as the tearing starts, since computing the principal stress on one vertex and computing the tearing path requires significantly less computation than computing principal stress on every vertex. The T-shirt scenario is more complex in terms of collision, with full collision with the underlying body, which explains the more important computation time.

Scenario	# Triangles	Collision	FEM	Tearing	Total
Cloth	1.4K	1.8	8.7	3.6/2.2	14.1/12.7
Banknote	1.9K	2	11	4.5/2.3	17.5/15.3
T-shirt	3.3K	4.5	14	6/2.5	24.5/21

Table 4.1 – Computation time, in ms, decomposed in three parts: collision detection computation, FEM deformation computation, and tearing-exclusive computation. Two times are provided for the tearing contribution, before and after tearing has started, with and without the principal stress computation for each vertex.

Feedback force during the tearing simulation can be seen in Figure 4.7, with the force feedback intensity along the tearing axis for both hands. As expected, it can be decomposed in several clear phases, first an elastic deformation. The object then starts tearing, and the tearing propagates with an almost stable force at a fraction of the maximal force. Both hands have similar force profile (with opposite direction), the small difference coming from the not completely symmetric tearing scenario.

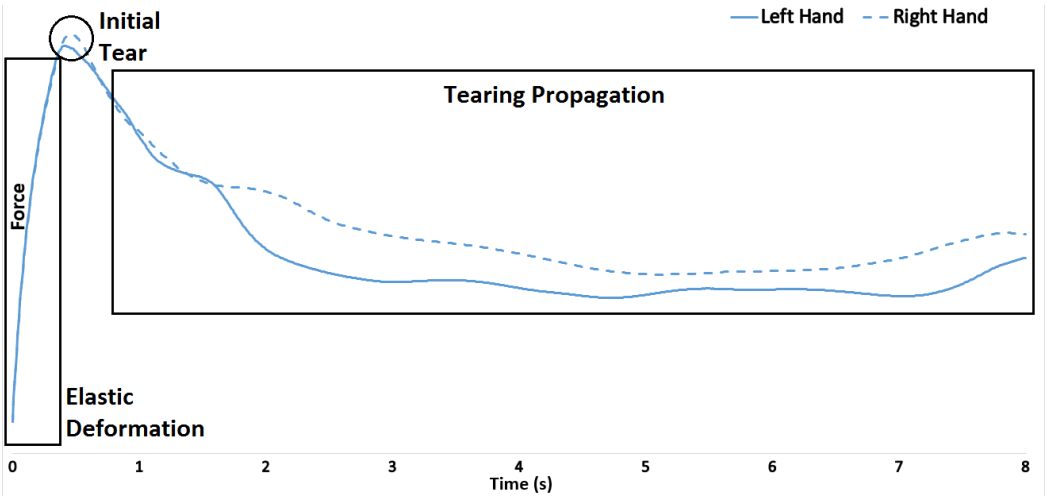


Figure 4.7 – Force feedback during the tearing process illustrating the 3 main phases for tearing simulation: initial elastic deformation, initial tearing, and tearing propagation. The force intensity along the horizontal axis is displayed for both hands.

Table 4.2 depicts the performances of the clustering methods for collision detection. To show the usefulness of our relative displacement measurement with cluster decomposition, we compared it with an absolute displacement measurement. The

computation time (in ms) is given for all 4 combinations of moving objects and displacement measurement. As expected, for non-moving objects, the performances are rather similar with both displacement measurement methods. But for moving objects, relative displacement measurement provides a great improvement in the performances of collision detection because our method is able to detect that the relative displacement between the t-shirt and the body is low even if they are both moving (which allow a higher usage of incremental methods in the collision detection).

Moving	Displacement Measurement	Mean Time	Deviation
no	absolute	3.65 ms	1.22
no	relative	3.37 ms	0.75
yes	absolute	11.76 ms	2.01
yes	relative	3.18 ms	0.46

Table 4.2 – Comparison of the performances of the collision detection method, with mobile and immobile objects, and using clusters or not. The mean computation time (in ms) over 400 simulation steps is provided, as well as the corresponding standard deviation.

7 Conclusion

In this chapter, we presented a fully-functional interaction pipeline enabling haptic tearing of deformable surfaces. It relies on both a highly efficient collision detection algorithm and an efficient FEM-based tearing simulation method. We proposed an improved method for collision detection of moving objects with little relative movement, such as a worn cloth, relying on the clusterization of the collision models. The tearing method is compatible with the use of multiple haptic devices simultaneously, in order to provide a realistic tearing experience. It has been implemented on two different examples illustrating potential applications of our method.

As of today, the physically-based simulation is performed on the CPU, which prevents the integration in large scale environments. While GPU-compatible, the collision detection was also performed on the CPU, since the small scale of our environments made the memory transfer cost between CPU and GPU too important compared to the corresponding gain. The method is also limited to surfacic elastic objects.

This method does however represent a step towards environments involving a extensive set of interactions through haptic means.

Study on haptic perception of stiffness in VR versus AR

5

Contents

1 Introduction

How are objects perceived in AR, from a haptic perspective? With the emergence of consumer-grade AR devices and the growing use of AR applications intended to provide better tools for the medical personnel, from training tools [Granados et al., 2017] to minimally-invasive surgery assistance [Nicolau et al., 2011] [Haouchine et al., 2013], such a question must be addressed. Haptic feedback provides important information on the organ that a doctor is interacting with, in order to know that the performed operation is correctly performed, or to detect an anomaly in the organ properties. In such domains, the accuracy of the feedback is therefore critical, since a wrong haptic feedback could potentially lead to a wrong diagnostic, or a misperformed surgical operation. In order to provide an accurate feedback, it is important to know how this feedback is perceived in an AR context.

In this context, after the previous simulation-based chapter aiming at simulating complex haptic interaction with objects undergoing topology changes, this contribution studies haptic stiffness perception, comparing the differences that might arise when changing the environment, from virtual to augmented. AR and VR, while relying on

similar technologies, provide a different kind of visual feedback, and vision is known to have a great influence over the haptic modality [Srinivasan et al., 1996], as explained in section 1.4. A main difference between the two is the presence, or not, of real objects in the field of vision of the user. It is however not clear how this difference can potentially influence the perception of the user. In other words, how different is perception in AR from perception in VR?

Visual perception in AR has been rather widely studied, taking into account several parameters such as the environment, the augmentation, the display device, or even the user [Kruijff et al., 2010]. Some biases in visual perception well documented in VR, such as distance underestimation, have also been observed in AR [Jones et al., 2008], with a lesser magnitude. However, though there exist previous studies on visual perception in AR and its difference with VR, there are actually very few studies on other sensory modalities, and in particular on the haptic sense.

Is haptic perception in AR different from haptic perception in VR? The presence of real objects in AR might indeed influence the way we interact with virtual objects and, eventually, the way we perceive them. In the end, the question we raise here is: does it feel the same when you touch an object in Augmented Reality or when you touch it in Virtual Reality?

In this chapter, we study how haptic perception of stiffness of a virtual object is influenced by displaying the scene in AR versus in VR. We conducted an experiment based on a HoloLens in which participants could interact with an object (a virtual piston) inside a real scene and inside a virtual reproduction of this scene. The participants were able to press on the virtual piston and perceive its stiffness using a force-feedback haptic device. They could successively compare the stiffness of the virtual piston in AR and in VR, with various levels of stiffness difference.

In the remainder of this chapter, we describe the protocol and apparatus of our experimental study in Section 2. The results obtained are presented in Section 3, followed by a discussion in Section 4. The chapter ends with a general conclusion in Section 5.

2 User study

This experiment aims at studying the potential influence of visual display, i.e. using Virtual Reality versus Augmented Reality, on the haptic perception of a virtual object (a piston). More specifically, we studied the influence of the nature of the visual surrounding of the piston (real or virtual) on its perceived stiffness. Participants had then to compare the stiffness of two pistons displayed sequentially. One of the piston was displayed in AR and the other one in VR (see Figure 5.5), in a counterbalanced order. **The reader is encouraged to look at the following video for a comprehensive description of the experimental apparatus and procedure.**

http://people.irisa.fr/Benoit.Le_Gouis/Videos/ISMAR_2017.mp4

2.1 Participants

12 participants (11 males, 1 female) took part in the experiment. They were aged from 20 to 29 (mean= 23.7, SD= 3.2). All of them were right-handed.

2.2 Experimental apparatus

The display of the virtual elements in AR and VR environments was achieved using a Microsoft HoloLens visual display[1]: a see-through HMD that can superimpose images on a portion of the field of view, with built-in tracking possibilities, displayed in Figure 5.1.

Figure 5.1 – The Microsoft Hololens AR display

In order to provide the participants with the same field of view in both AR and VR environments using the HoloLens, the peripheral field of view was hidden using a mask made of a piece of tissue with two rectangular holes for the eyes and attached to the HoloLens (see Figure 5.4). Thus, the remaining field of view corresponds to the on-boarded screens of the HoloLens and could be fully superimposed with the virtual scene.

The experimental setup is then based on a visual scene composed of a cardboard box containing several objects with simple shapes: a glue stick, a Rubik's cube, a red clown nose and three violet dice (see Figure 5.3). These objects are easily recognizable by most people, with known dimensions, and thus provide relevant information about the size of all objects inside the virtual environment, including the object that was interacted with. The background for the interaction needed to be textured as to not influence the distance perception, and the texture had to contain no pattern that could also have influenced the perception. The inner faces of the box were thus covered with printed sheets of paper displaying colored random dots. The lighting was carefully

[1] www.microsoft.com/microsoft-HoloLens

provided by two LED projectors as to: (1) fully illuminate the scene, and (2) provide sufficient light levels for the real scene to be brightly lit, but not too strong for the HoloLens to be able to occlude efficiently the real environment in VR, as depicted in Figure 5.2.

Figure 5.2 – The complete setup used for the experiment, including a careful handling of the lighting conditions.

Participants were comfortably seated 2-meters in front of the cardboard box, allowing them to see all the scene at once with the HoloLens, yet with sufficient details (see Figure 5.4).

The scene was entirely reproduced in a faith-full manner in VR, including: the cardboard box, the objects, the front wall, the table and the lighting conditions. In the VR environment, due to the good occlusion capacities of the HoloLens and the careful handling of the lighting, the real scene was almost invisible. In order for the virtual box to be displayed in front of the real cardboard box, it has been made slightly smaller.

A virtual piston was then superimposed on the real scene in AR, or integrated to the virtual environment, as depicted in Figure 5.5.

In order to avoid any bias linked with a potential difference in the display method and associated field of view, we decided to create a VR environment by using a AR device, and filling the entire field of view with a virtual environment, which would be an accurate copy of the real world at the same location. With this method, the piston is exactly the same in the AR environment and in the VR environment, and only the surrounding elements are changed. Having two different display modalities would also require a participant to switch constantly, which would first greatly affect the performances with a long delay between the two compared stimuli and also make the

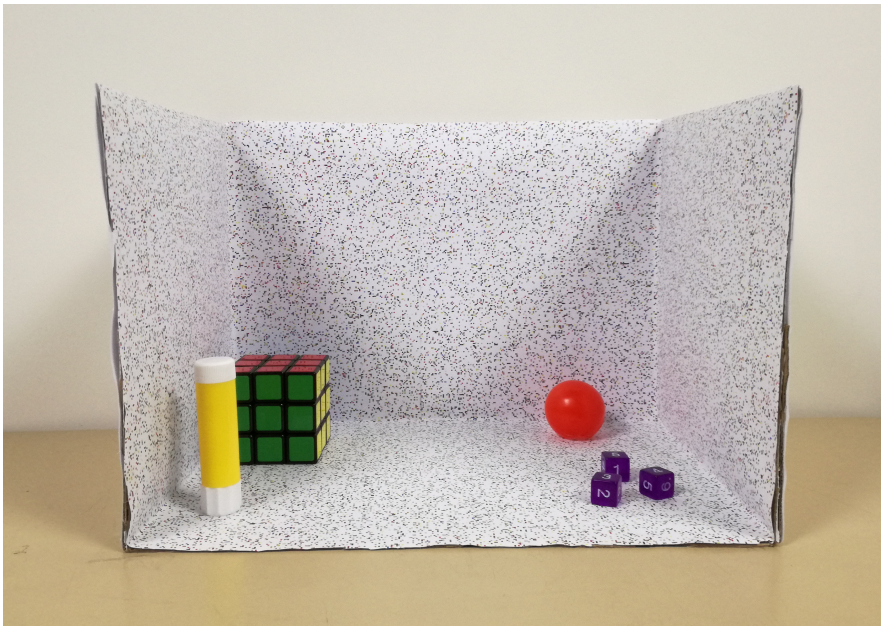


Figure 5.3 – Close-up of the scene. A cardboard box with a colored texture (random colored dots) contains several casual objects: a yellow glue stick, a Rubik's cube, a red clown nose, and three violet dice.

experiment really long, which might induce fatigue that could also impair the results. Having the same display also reduces the chance of having a different perceived color – and hopefully depth – of the piston, which might influence the performance as well.

Participants could interact with the virtual piston using a haptic force-feedback device (Falcon, Novint company). They manipulated a 3D cursor (represented as a 3D blue sphere) along 3 degrees of freedom, with a 1:1 mapping between the motion of the haptic device extremity and the motion of the 3D cursor. Once the 3D cursor was in contact with the virtual piston the participants could exert pressure on it. The stiffness of the piston was then rendered using the force-feedback and simulating a pure spring along the vertical axis. The Falcon device was positioned sideways in order to ensure higher forces and a more homogeneous haptic manipulation workspace. The haptic rendering was handled by a remote computer using CHAI3D software API [2], and the position of the haptic device was sent to the HoloLens using WiFi which is the preferred means of communication of the HoloLens.

Participants also used a numerical pad attached using a band to their left leg with two keys labeled *"1"* and *"2"*, in order to answer the questions in a comfortable manner with their left hand.

The choice of the non co-located interaction was motivated by two major constraints, first, the field of view of the HoloLens and second, the integration of the haptic device and the hand of the participants in the VR condition. The proposed setup enables the entire scene to be visible from the HoloLens and avoids the integration of the haptic device and the participant's hand. Although having a non co-located interaction might have an influence on the haptic perception [Congedo et al., 2006], this effect

[2] www.chai3d.org

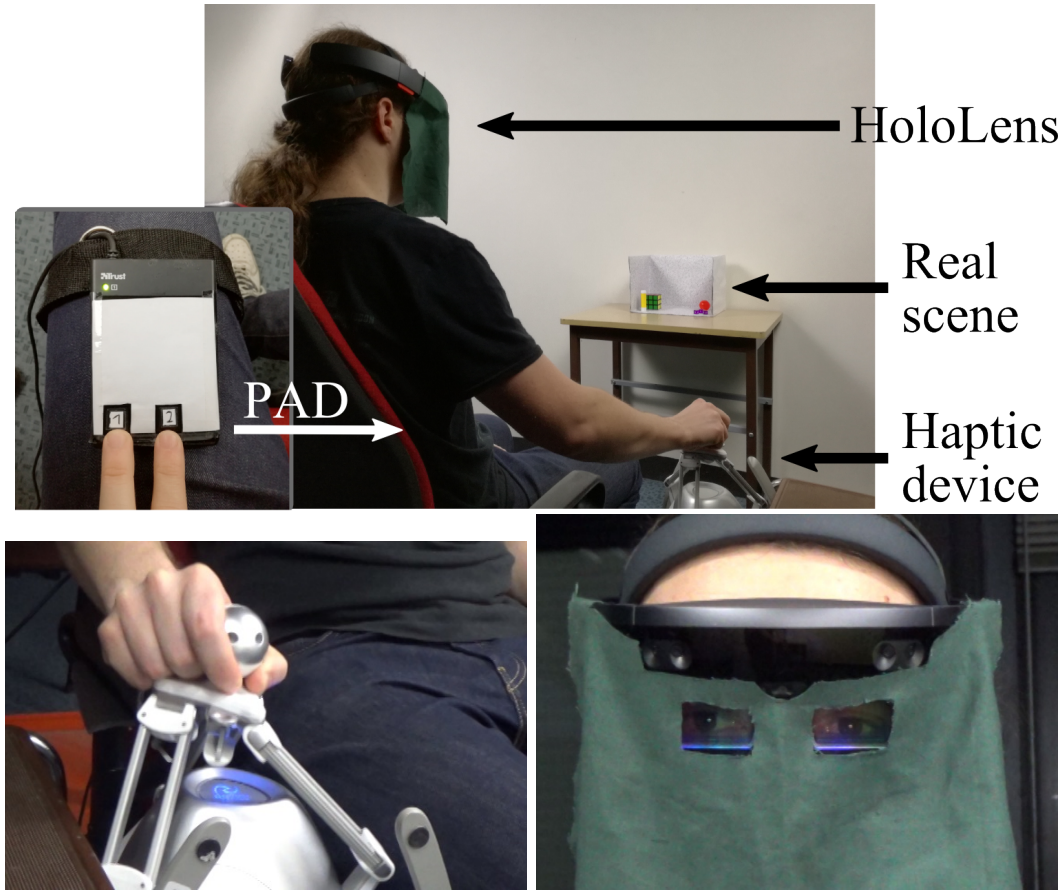


Figure 5.4 – Experimental setup. The participant is seated in a comfortable chair wearing a HoloLens device and uses a pad to answer which of two virtual pistons was the stiffest (top). He interacts with a virtual piston using a Novint Falcon haptic device located at his side (bottom-left). A mask (bottom-right) with two holes for the eyes and made of tissue is fixed on the HoloLens so to hide the peripheral field of view which cannot be augmented by the HoloLens.

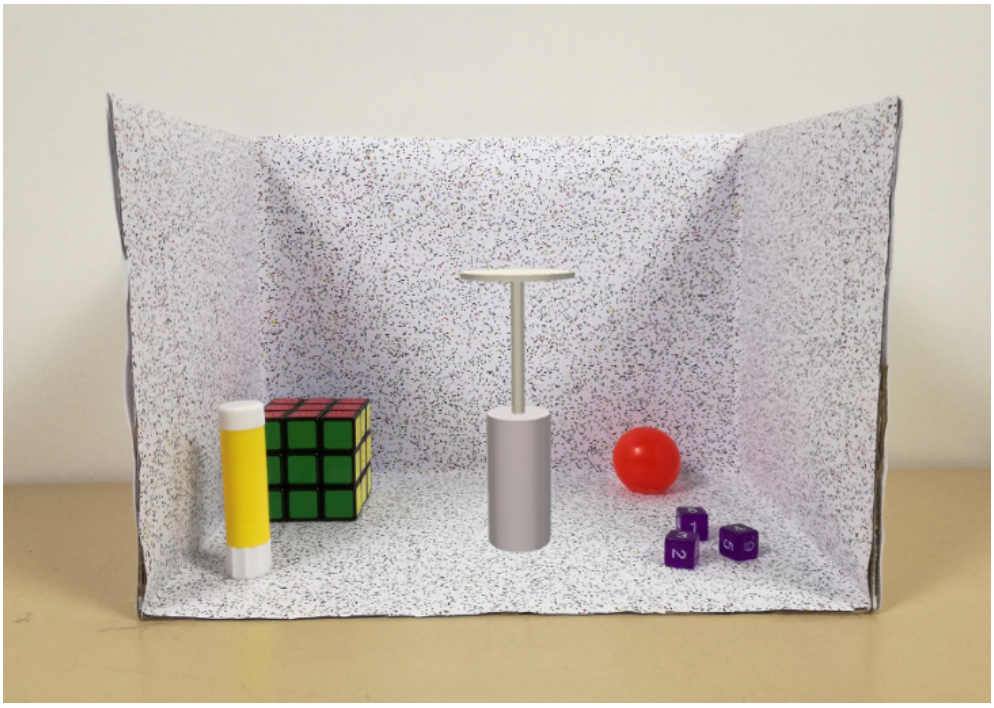
would equally affect the perception in both AR and VR conditions and should thus not drastically alter the experiment results.

2.3 Conditions and Plan

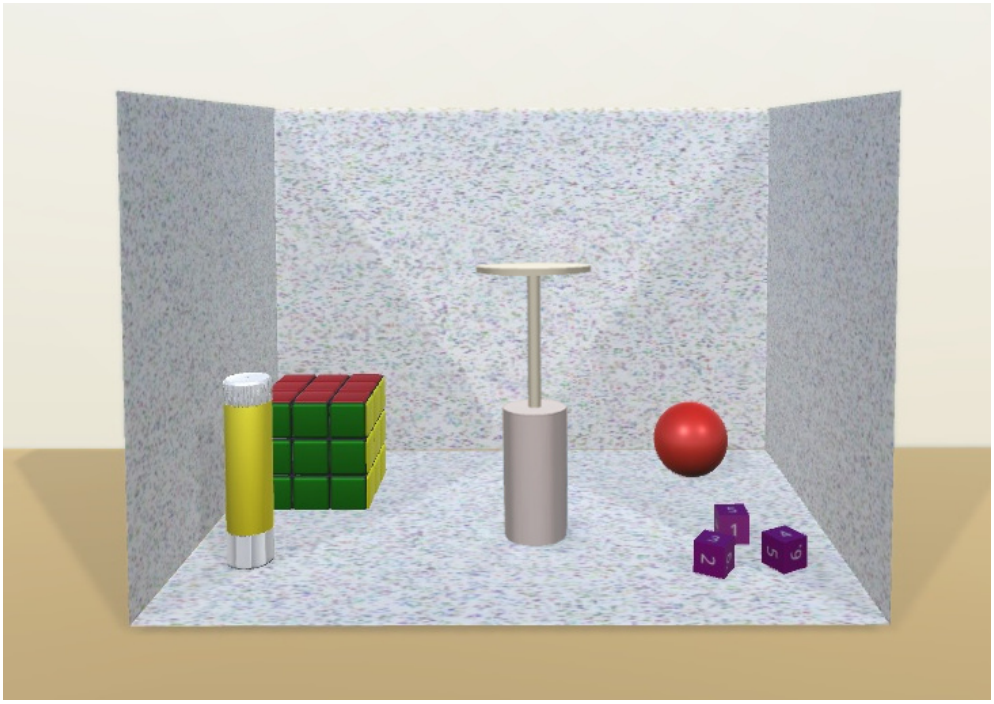
There were two environment conditions related to the visual display. The **AR condition** corresponds to the use of an augmented reality display mode, whereas the **VR condition** corresponds to the use of a virtual reality display mode.

Then, two specific conditions are considered:

- **C1** is the visual condition of the reference piston. (AR reference) means that the reference piston is displayed in AR, and (VR reference) means that the reference piston is displayed in VR. The value of the stiffness of the reference condition was set in both cases to 0.11 N.mm^{-1} after preliminary testing.



(a) Virtual piston in AR.



(b) Virtual piston in VR.

Figure 5.5 – Experimental conditions. (a) AR environment with a virtual piston superimposed inside the real cardboard box. (b) VR environment with the same virtual piston located inside the virtual scene.

- **C2** is the stiffness value of the comparison piston. Five possible values were chosen after preliminary testing, corresponding to the following five differences: −16%, −8%, 0%, +8% and +16% compare to the reference stiffness.

The order of presentation of the two pistons and their display environment was counterbalanced to avoid any order effect [Ziemer et al., 2009]: every couple of pistons was presented in all orders (AR first/VR first, reference first/comparison first).

Thus, participants were presented with 100 trials, divided in 5 blocks of 20 trials. Each block of 20 trials presented a set of couples of pistons made of: 2 stiffness references (**C1**) × 5 stiffness values (**C2**) × 2 presentation orders (AR then VR, or VR then AR) = 20 trials), in a different randomized order for each block.

2.4 Procedure

Participants started by filling out a short form. After verbal explanations, they performed 5 training trials during which they could get used to the experiment procedure. Then, the participants were presented with the set of 100 trials. Every trial was processing as follows (see also 5.6) :

A real scene (AR condition) or virtual scene (VR condition) was displayed (see Figure 5.5), all including a virtual piston and a 3D cursor (blue sphere).

- A red cylinder located over the piston represented the starting position volume, as depicted in Figure 5.6a.The participants had to reach and remain in the starting position volume for 1 s before being able to interact with the piston.

- After that delay, the cylinder turned green (Figure 5.6b), and the participant could interact with the piston for 3 s, as seen in Figure 5.6c.

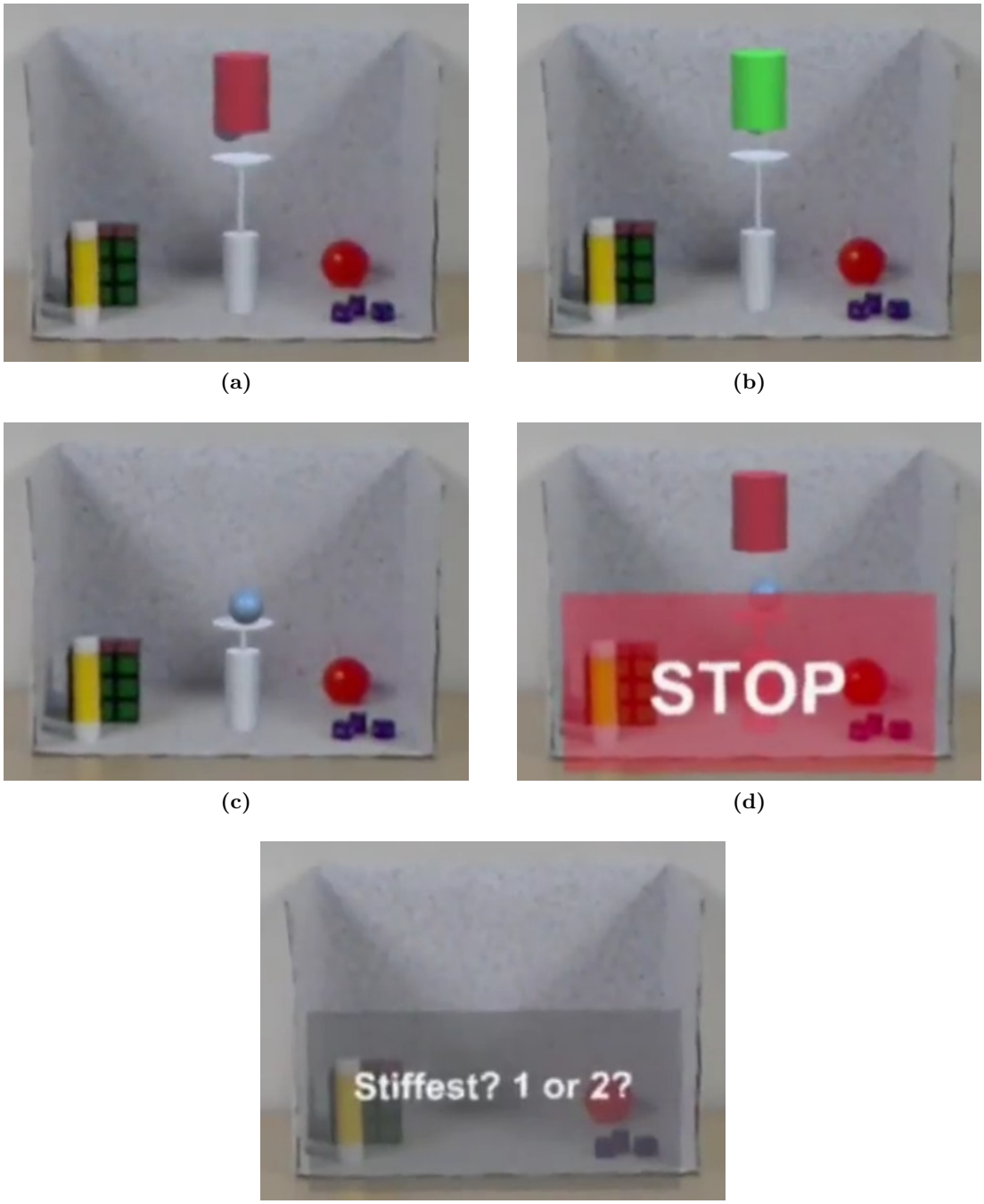
- At the end of the exploration time, a stop message and panel popped in front of the scene, and the red cylinder reappeared (Figure 5.6d).

- When the participants reached again the red cylinder, the condition (AR or VR) changed, as well as the stiffness of the second piston. The participant still needed to stay inside the red cylinder (starting position volume) for 1 s before being able to interact again with the second piston.

- After 3 s of interaction with the second spring, the stop panel reappeared, and after reaching the starting position volume for the third time, the participant was presented with the response panel asking which was the stiffest pistons (1 or 2), as shown in Figure 5.6e. The participants could enter their answer using the pad attached to their left leg, as displayed in Figure 5.4.

- Once the answer of the participant was recorded the next trial started.

After each block of 20 trials, a break was proposed to the participant.



(a) **(b)** **(c)** **(d)** **(e)**

Figure 5.6 – Experimental procedure (displayed here in the AR condition). (a) A red cylinder displays the starting position to reach with the manipulated cursor (blue sphere). (b) The cylinder turns to green to inform the participant that he/she can start evaluating the stiffness of the first piston. (c) The participant can press on the piston using the manipulated cursor. (d) A stop sign and panel indicates that the evaluation time is over. The same sequence (a-b-c) is then proposed in the second condition (VR here). Then, (e) The participant must answer, ie report which piston is the stiffer. Pictures were captured using HoloLens' camera.

As the HoloLens communicated with the computer managing the Falcon and pad using Wi-Fi, connection problems between the HoloLens and the computer could occur. These problems caused a noticeable drop in the update frequency for the displayed position of the haptic device, down to 3 Hz. Participants were asked to verbally warn when such problems happened, in order for the corresponding trial to be discarded from the analysis. Around 3% of the trials were discarded that way.

2.5 Collected data

For each couple of pistons, we collected 5 objective measures:

- **Om1 Participant's answer** is the piston (1^{st} or 2^{nd}) which was reported by the participant as the stiffer one.
- **Om2 Response time** corresponds to the elapsed time between the end of the evaluation of the second piston and the moment the participant entered his/her answer.
- **Om3 Displacement quantity** corresponds to the sum of every vertical displacement (absolute value, in meters) of the haptic device when in contact with the piston during the interaction time. This measure was recorded separately for the two presented pistons.
- **Om4 Force** corresponds to the average force (in N) the participants received from the device over the interaction time. This measure was recorded separately for the two presented pistons.

Participants also completed a subjective questionnaire, detailed (in French) in Appendix C, at the end of the experiment. Each question of this questionnaire was answered using a 7-items Likert scale:

- **Sm1** *"The piston seemed real in augmented reality."*
- **Sm2** *"The piston seemed real in virtual reality."*
- **Sm3** *"I did not see the real environment when the scene was entirely virtual"*. This question was asked to evaluate the quality of the occlusion of the real scene in the VR condition.
- **Sm4** *"Except for their real/virtual aspect, I did not notice any difference between the augmented and the virtual scenes"*. This question was asked to evaluate the correctness of the reproduction of the virtual scene compared to the real scene.
- **Sm5** *"After the experiment, I felt visual fatigue."*
- **Sm6** *"After the experiment, I felt haptic fatigue."*

We also asked an open question to the participants **Sm7** *"Do you think that the real environment influenced your haptic perception of the virtual piston? If so, how?"*. This question was asked to get a subjective feedback concerning the possible influence of the type of environment on the stiffness perception.

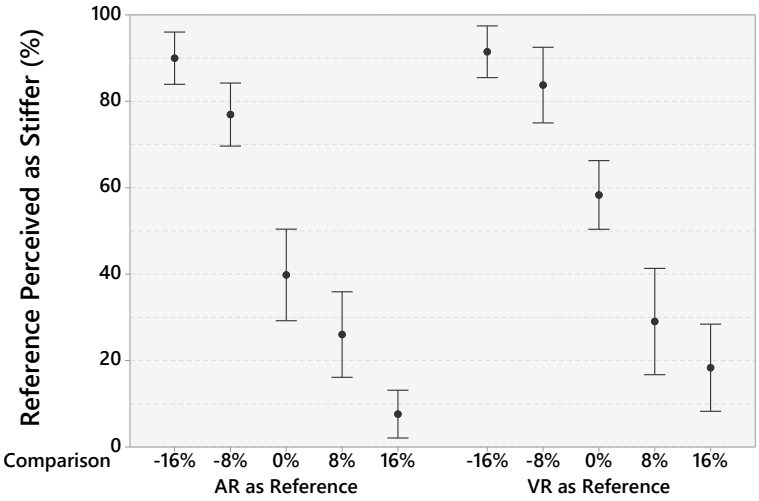


Figure 5.7 – Confidence intervals (95%) for the participants answer "the comparison spring is stiffer than the reference one" for the five stiffness conditions and the two reference environment conditions.

3 Results

3.1 Recognition Accuracy

In order to analyze the participants' answers (**Om1**) we pooled the data for all repetitions (no ordering effects were found). For each combination of factors, we computed the percentage of the answers in which the comparison object (spring) was perceived to be stiffer than the reference object, then we performed a two-way ANOVA analysis considering the nature of the reference environment (**C1**) and the stiffness of the comparison object (**C2**) (see Figure 5.7).

The ANOVA showed a main effect for **C1** ($F_{1,11} = 15.72$; $p < 0.01$; $\eta_p^2 = 0.59$). Post-hoc tests showed that when the environment was virtual, the reference object was significantly considered stiffer $M = 0.51$; $SD = 0.33$ than when the environment was real $M = 0.44$; $SD = 0.32$. This refutes the null hypothesis, which is "there is no difference between the two conditions".

In addition, we also observed a main effect for **C2** on the stiffness of the virtual object ($F_{4,44} = 100.48$; $p < 0.001$; $\eta_p^2 = 0.90$). As expected, as the stiffness of the comparison object increases, it is considered to be stiffer than the reference object (see Figure 5.7). The ANOVA did not show any interaction effect ($F_{4,44} = 1.95$; $p = 0.119$; $\eta_p^2 = 0.15$).

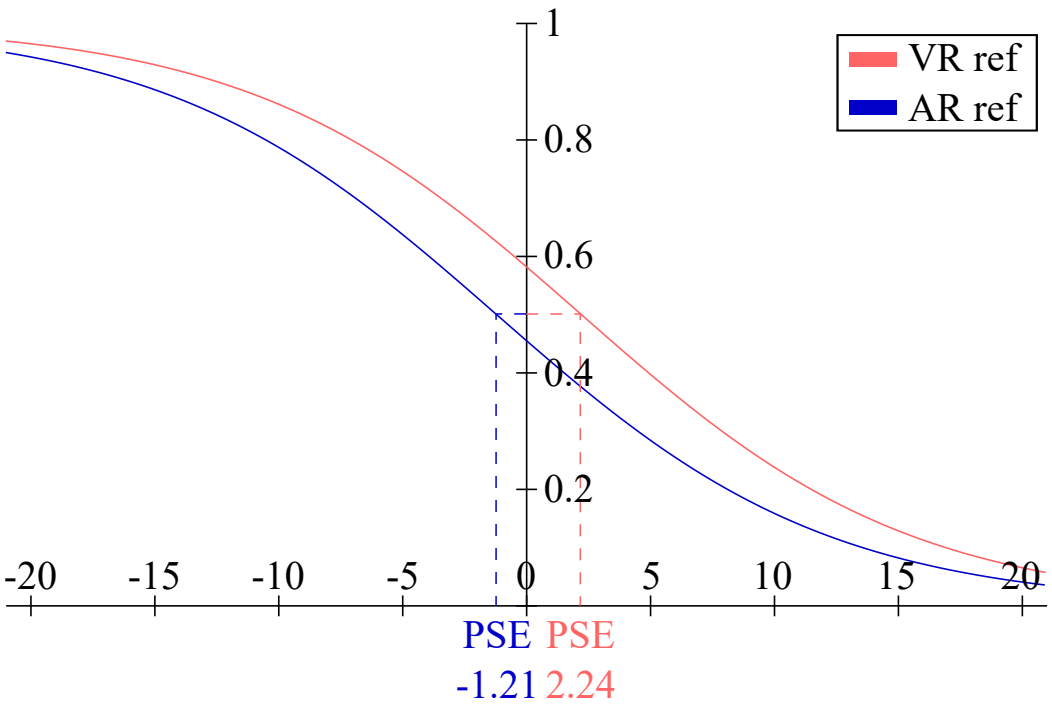


Figure 5.8 – AR versus VR psychometric curves. The red (resp. blue) curve shows the psychometric curve with VR (resp. AR) as a stiffness reference. The corresponding Point of Subjective Equality (PSE) is displayed for each condition. Curves were plotted using the dedicated psignifit software.

Due to the significance of **C1**, we further analyzed the recognition accuracy by fitting psychometric curves (see Equation 5.1) to the data based on the question: is the comparison object stiffer? We computed the curve for each level of **C1** using the dedicated psignifit software[3] (see Figure 5.8).

$$f(x) = \frac{1}{1 + \mathrm{e}^{-\frac{x-\alpha}{\beta}}} \tag{5.1}$$

The obtained coefficients were $\alpha = 2.24$ ($CI = [0.03, 4.06]$) and $\beta = -6.69$ ($CI = [-8.09, -5.18]$) for the VR reference condition and $\alpha = -1.21$ ($CI = [-3.18, 0.6]$) and $\beta = -6.72$ ($CI = [-8.85, -5.84]$) for the AR reference condition. Such α coefficient determines a Point of Subjective Equality (PSE) which represents the value in which both pistons are considered to be equivalent (e.g. 50% of chance to choose one or another). The lower value of the PSE for the VR reference condition supports the ANOVA results on the significance of **C1**.

The corresponding JND values were 11.09 ($CI = [7.89, 14.44]$) for the VR condition and 11.14 ($CI = [9.09, 14.9]$) for the AR condition.

[3] https://github.com/wichmann-lab/psignifit/blob/master/psignifit.m

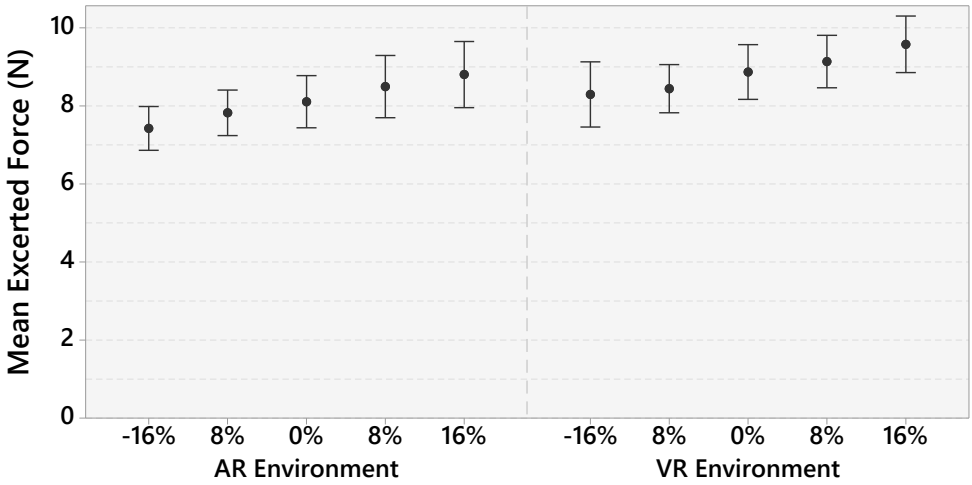


Figure 5.9 – Confidence intervals (95%) for the mean exerted force for the five stiffness conditions and the two environment conditions.

3.2 Remaining objective measures

- **Response Time:**

  We evaluated the influence of **C1** and **C2** on the time participants needed to answer (**Om2**). The two-way ANOVA **C1** and **C2** vs. answering time did not show any significant effect. On average, participants required $M = 1.75s$; $SD = 1.15s$ to respond.

- **Spring Displacement:**

  Regarding the total displacement applied (**Om3**), the two-way ANOVA **C1** and **C2** vs. displacement, showed a main effect on **C1** ($F_{1,11} = 6.60$; $p < 0.05$; $\eta_p^2 = 0.37$). Yet, the relevance of this significance is limited due to the mean differences and the data variability: VR condition $M = 0.217cm$; $SD = 0.12cm$, AR condition $M = 0.225cm$; $SD = 0.13cm$. No main effect on **C2** ($F_{4,44} = 1.44$; $p = 0.236$; $\eta_p^2 = 0.11$) nor interaction effects were found ($F_{4,44} = 2.00$; $p = 0.111$; $\eta_p^2 = 0.15$).

- **Force Exertion:**

  Regarding the exerted force (**Om4**) (see Figure 5.9), the two-way ANOVA **C1** and **C2** vs. force, showed a main effect on **C1** ($F_{1,11} = 53.52$; $p < 0.001$; $\eta_p^2 = 0.83$) and on **C2** ($F_{4,44} = 35.82$; $p < 0.001$; $\eta_p^2 = 0.77$). Post-hoc tests showed that participants significantly exerted more force in the VR condition ($M = 8.86N$; $SD = 1.18N$) than in the AR condition ($M = 8.13N$; $SD = 1.17N$) and that the exerted force increased with the stiffness of the spring. No interaction effects were found ($F_{4,44} = 0.30$; $p = 0.877$; $\eta_p^2 = 0.03$).

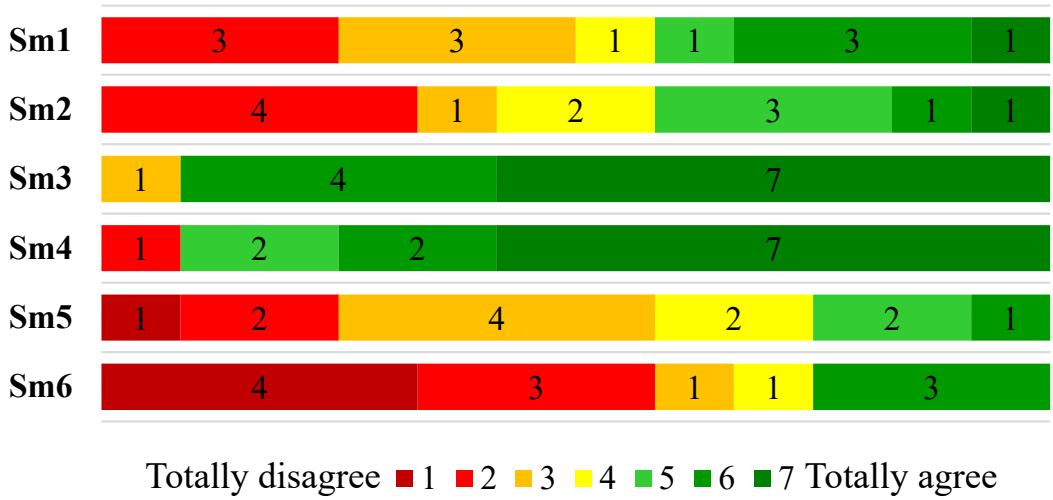


Figure 5.10 – Subjective questionnaire results. Each line corresponds to the answer of the participants for a subjective measure, evaluated on a 7-point Likert scale. Green colors correspond to positive answers. Red colors correspond to negative answers.

3.3 Subjective answers

Figure 5.10 presents the answers collected through our subjective questionnaire (7-point Likert scale). Regarding the appearance of the virtual piston, participants reported that the virtual piston barely seemed real in AR (**Sm1**, $M = 4.08$; $SD = 1.83$) and in VR (**Sm2**, $M = 3.92$; $SD = 1.73$). A Student test showed this difference was not significant ($p = 0.55$, $Qobs = 0.62$).

Regarding the quality of the AR display, 11 participants reported that the virtual scene correctly occluded the real environment, 7 of which gave the maximal rank (**Sm3**, $M = 6.33$; $SD = 1.15$). Except for one participant, participants did not perceive any strong difference between the real and the virtual scenes, 7 of them giving the maximal rank (**Sm4**, $M = 6.08$; $SD = 1.51$). Two participants reported difference in luminosity between the two scenes, in favor of the virtual scene.

Five participants reported a positive or neutral visual fatigue during the experiment (**Sm5**, $M = 3.42$; $SD = 1.44$). One participant reported that the lighting in VR was tiring. Participants reported overall medium levels of fatigue (**Sm6**, $M = 2.92$; $SD = 2.07$), and three of them reported higher levels of fatigue. Concerning the last open question, 8 participants reported they did not think the environment influenced their haptic perception (**Sm7**). One participant reported a slight difference because of head orientation. Two participants reported the piston felt softer in VR. One participant reported the piston felt softer in AR.

4 Discussion

The results of our psychophysical study show a difference between the stiffness perceived in augmented reality and the stiffness perceived in virtual reality. The virtual piston was significantly more often perceived as stiffer in the VR condition than in the AR

condition. In particular, given an equal stiffness between the two pistons in AR and VR, the participants on average reported that the piston was stiffer in the VR condition around 60% of the time. Moreover, the two Points of Subjective Equality (PSE) that we could compute (between a reference piston tested in one condition and a comparison piston tested in the other condition) are different, suggesting a perceptual offset of 3.45% on average. Thus, taken together, our results suggest a psychological effect or bias, as if the piston tested in a purely virtual environment feels "stiffer", and the same piston tested in an augmented (real) environment feels “softer”.

The JND values found in our experiment are around 11%. This value is smaller than the one usually found in the literature for stiffness discrimination (between 15% and 22%), and closer to the JND found for force discrimination (10%). However, contrary to the participants in these studies, participants in our study could see the object, which makes the discrimination task easier.

From the subjective questionnaires, one can notice that the quality of our VR scene seems to be well appreciated, and estimated as a convincing reproduction of the real scene:

- The participants have indeed found that the real scene was well occluded by the virtual scene in the VR condition (**Sm3**).
- They found very little difference between the AR and the VR scenes (**Sm4**).
- The participants reported low levels of visual (**Sm5**) or haptic (**Sm6**) fatigue after the experiment.

We performed an additional analysis comparing recognition rates and answering times between the first and last blocks of the experiment. A Wilcoxon signed-rank test showed no difference, suggesting that – even though little – the reported fatigue did not influence the collected measures.

Surprisingly, almost all participants reported that the type of display (AR versus VR) did not influence their haptic perception (**Sm7**). This suggests that the participants were not aware at all that the visual condition had an influence on their answer. However, participant 3 reported that "The piston felt stiffer in VR because all elements are congruent, i.e. they are all virtual”, which is in line with our hypothesis.

Another interesting observation relates to the measures of force and displacement applied on the virtual piston in the two conditions.

- There was no difference found in terms of quantity of displacement applied on the piston between the VR and AR conditions (**Om3**).
- However, the participants received 11% more force in the VR condition compare to AR (**Om4**).

This means that the participants applied the same quantity of movement and probably kept on constantly applying oscillating pressures up and down. But they stopped their motion earlier in the AR condition and went "deeper" in the piston in the VR condition. As a result, they exerted and received more force when the scene was entirely virtual. This change in the exploration strategy could thus also explain the fact that the virtual piston in VR was perceived as stiffer as already observed in [Chen, 2017] and

[Endo, 2016]. Participants tend to press harder on objects perceived as stiffer, which could explain the difference in interaction strategy, but does not explain why the VR piston is perceived as stiffer. Another interpretation could here be that the participants felt "safer" in the virtual condition and/or "less confident" in the AR condition. In any case, this surprising difference in haptic interaction strategy – the fact that there is a greater motor involvement (and higher forces exertion) in the VR environment – calls for further behavioral studies.

One of the possible perceptual biases came from the fact that the virtual box is smaller than the actual cardboard box, a smaller size required in order to have the virtual box display on top of the real one in the VR setup. This difference in perception could have influenced the perception, with an effect close to the one used in pseudo-haptic applications. In order to ensure that our found psychophysical biases could not be only explained by this, we conducted a smaller experiment in which the scene did not contain the box, only the table, the wall and the objects at the same location. This experiment has been performed on 4 subjects. The obtained coefficients were $\alpha = 3.78$ ($CI = [-1.02, 6.98]$) and $\beta = -10.55$ ($CI = [-17.64, -7.75]$) for the VR reference condition and $\alpha = -5.18$ ($CI = [-8.87, -1.3]$) and $\beta = -9.43$ ($CI = [-16.12, -7.13]$) for the AR reference condition. While the experiment sample is quite small, the magnitude of the found difference indicates that the smaller box is not responsible for most of the found psychophysical bias, if responsible at all.

5 Conclusion

We studied haptic perception in augmented reality versus in virtual reality. We designed an experimental setup based on a Microsoft HoloLens visual display and a force-feedback device. Participants could press on a virtual piston in either in an AR or in a VR environment, and compare their stiffness.

The results of our psychophysical study show that the participants have perceived the virtual piston as “stiffer” in the virtual environment than in the augmented environment. In the case of equivalent stiffness between AR and VR, participants chose the VR piston as the stiffer one in 60% of cases. We found that the forces exerted by the participants on the virtual piston were higher in virtual reality than in augmented reality, suggesting different exploration strategies. Moreover, a participant reported that the virtual piston appeared “more real” in the virtual environment than in the augmented environment as it was more congruent with its environment.

Taken together our results suggest that haptic perception of virtual objects is different in augmented reality compare to virtual reality. In particular, they suggest a new psychological phenomenon: a bias in haptic perception making virtual objects feel "softer" in augmented environments compare to purely virtual environments.

Conclusion

6

This manuscript, entitled **Contribution to the study of haptic perception and rendering of deformable virtual objects** discussed work on complex haptic interaction with deformable objects, as well as results on the effect of the display technology for virtual objects on haptic perception.

The first contribution showed how this computation complexity could be reduced without perceptual difference in the case of heterogeneous objects, by changing the elasticity distribution inside the object as to simplify the geometric coarsening process. The second contribution enabled more complex haptic interaction with deformable surfaces with a pipeline for bimanual haptic tearing, including efficient collision detection and FEM-based simulation for tearing phenomena. The third contribution aims at better understanding the phenomena influencing haptic perception by studying the effect of the environment, virtual or augmented, on haptic stiffness perception. These results pave the way for promising work on the design of adaptive environments and haptic interaction with them, as well as studies in haptic perception, in order to make sure that the adaptiveness of the environments comes at a minimal perceptual cost.

1 Elasticity aggregation for geometric multi-resolution on heterogeneous objects

In Chapter 3, we presented a cluster-based method for the haptic interaction with multi-resolution heterogeneous objects. While many methods exist to handle heterogeneity for geometric multi-resolution, none consider the perception with the obtained low-resolution object. We proposed a method that modifies the elasticity distribution inside the high-resolution as to 1) create large homogeneous volumes that can easily be geometrically coarsened, and 2) take into account the initial distribution in order to have similar forces during the interaction with the original and the simplified object. This is performed by grouping tetrahedra of similar elasticity and homogenizing them into an average elasticity, in order to create large homogeneous volumes. Grouping the tetrahedra is performed with spatial clustering to know what tetrahedra should be grouped together, and tetrahedra from the same cluster are grouped around the barycenter, taking into account both the proximity to the barycenter to have a large volume without thin patterns and the position of the tetrahedra before grouping to take the original distribution into account. We tested our method on various elasticity distribution, during the interaction with a cube of three different resolutions. Results

show that for most distributions, the simplification allows to have a similar force to the original, where the naive solution to homogenize the cube would fail in most cases. While currently limited to simple shapes and interaction this method could be extended with a domain decomposition of the object, simplifying only the parts that are not in direct contact with the haptic probe.

2 Haptic tearing of deformable surface

In Chapter 4, we presented a pipeline for bimanual haptic tearing of deformable surfaces. Tearing is an important aspect in the simulation of deformable surfaces, and must thus be taken into account during haptic interaction. The pipeline is composed of efficient collision detection for deformable surfaces, FEM-based simulation with special optimization for the tearing simulation, and an asynchronous haptic rendering scheme for bimanual interaction. The collision detection algorithm decomposes the deformable surface into patches as to only recompute the narrow phase collision detection algorithm when there has been enough relative displacement between the patch and the colliding element. This method allows improved performance in the case of low relative displacement between different parts of the simulation, such as worn cloth. Simulation of the deformable surface is performed using FEM, with special handling of the tearing simulation. Tearing occurs at places where the local stress reaches a certain threshold. Tear propagation is optimized as to improve computation performances. Bimanual haptic tearing is enabled through an asynchronous rendering scheme.

3 Comparison of haptic stiffness perception in AR vs VR

In Chapter 5, we presented the results from a psychophysical study on the influence of the environment – AR or VR – on haptic stiffness perception. In this experiment, we created a virtual environment containing simple objects, and a reproduction of this scene with the corresponding real objects in a cardboard box. Using a AR headset, the Hololens, participants could interact with a virtual piston embedded in each of the environment, creating a complete virtual environment in one case, and an augmented environment in the other case. The interaction was performed using a haptic device. Both conditions were presented subsequently, with various stiffness differences. These results suggest a psychophysical bias in perception, a piston in VR is perceived as stiffer than the same piston in AR, with participants choosing the VR piston over the RA piston as the stiffer one 60% of the time in case of equal stiffness. Participants would on average push harder on the virtual piston in the VR environment, suggesting a different interaction strategy between the two environments. This different strategy might partially be explained by the different levels of integration of the piston compared to its surroundings, the virtual piston fitting more naturally inside the virtual environment than to the real reconstitution. This difference in strategy calls for further investigation on the conditions under which there is a perceptual offset.

4 Future work

4.1 Geometric multiresolution on heterogeneous objects

Future work on our approach could first include the possibility to provide haptic rendering with different tools. As of today, the elasticity is modified over the entire object, and local interactions on a small part of the object are greatly impacted by our method. In order to provide a similar interaction over the different mesh resolutions, our method requires a tool that interacts with a significant part of the object at once, for example a large spatula pressing a great part of the top surface. This limitation could be overcome by performing a domain decomposition on the object, and our method and the corresponding geometric coarsening could be applied on all domains except for the one in direct interaction, as depicted in Figure 6.1. Since most of the computation in our method is performed offline, the domain geometry can be changed to reflect a change in the interaction location, enabling interaction with the entire object.

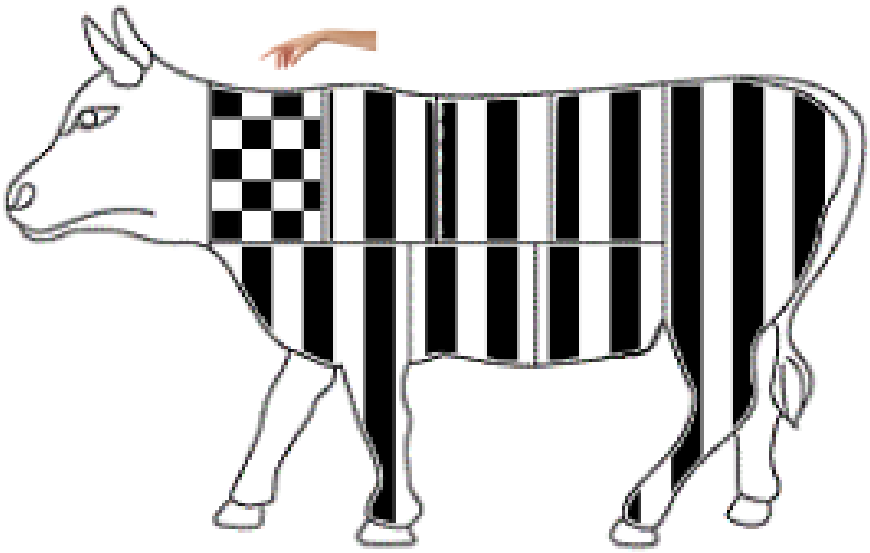


Figure 6.1 – An object is decomposed into domains, and all domains except for the one in direct interaction are coarsened

It would provide the accurate deformation on the surface with the precise mesh, and the coarse interaction between domains represents a good approximation of the force from the remaining of the object, with the border between domains representing a large surface with a size that is significant compared to the size of the domain. With such a decomposition, topology modifications such as cutting gestures could thus broaden the spectrum of possible applications, with domains reverting to non-modified state as soon as they are in the cutting path. Further validation could also include a thorough comparison with the different methods from the literature, both for computation time performances and for the perceived interaction force. A comparison with different spatial clustering methods could also prove to be interesting. Moreover, as of today, the optimal simplification level relies on a prior knowledge of the elasticity

distribution inside the object, and an analytical tool computing the optimal clustering level for a given object based on its distribution could be of interest. Furthermore, while depending on well-studied results on perception, we did not formally validate our method with a user study, which could evaluate the haptic perception of our approach.

Thanks to the performance gain obtained by geometric coarsening at no perceptual cost, our method allows the interaction with a great number of heterogeneous objects. This could lead to potential applications in the entertainment field, with haptic interaction with a large environment, or to virtual prototyping applications.

4.2 Tearing simulation

Future work could include the tearing simulation of objects undergoing more complex phenomena such as plastic deformation or anisotropy. Since the collision detection is already compatible for GPU computation, enabling GPU for the physically-based simulation as well could significantly improve performances. Memory transfer cost between GPU and CPU with scenes of our scale make it not worthy to include GPU computation, but a full GPU integration of the simulation could allow the simulation of much larger environments with a great quantity of surfaces that could be chosen and torn in an interactive manner. Concerning collision detection, a dynamic re-computation of the cluster distribution could be performed when the cluster is separated into several non-connected pieces, thus improving the computation time for collision detection. Finally, an extension of the method to volume objects could broaden the spectrum of possible applications.

4.3 Stiffness perception in AR

Future work is necessary to deeper qualify how and why people have different exploration strategies, different ways of interacting, and different final haptic perception in such virtual versus augmented versus real environments. It provides with exciting future work on that phenomenon, in order be able to estimate the bias under various scene configurations, that might have an impact on the results.

Another aspect to study in future work is the impact of the realism of the scene on the perceived stiffness. While this realism is in part limited by technological limitations, a thorough investigation of the impact of the simulation- and rendering-dependent level of realism (color similarity, inter-objects reflexion, etc) on stiffness perception could indeed lead to a deeper understanding of the influence of the integration of the virtual object in an augmented environment on the haptic perception.

5 Long-term perspectives

5.1 Towards full perception-based adaptivity in a virtual environment

In the coming years, there should be an increasing need in the simulation of always more complex and large environments, led in parts by the entertainment industry, always pushing the technology limits with the size and interactivity of the virtual

environments in video games or simulators, and also for training or prototyping. This increase in the need of computation resources can be met in parts by the increase of computer power, yet there should remain an important need for efficient methods that guarantee the simulation and interaction quality.

While the work presented in this PhD represents a first step towards more perception-based adaptive environments, there is still a great potential of improvement to be done, especially from the haptic point-of-view. In order to reach that point, many other aspects must be taken into account, from the entirety of the interaction loop (collision detection and response with a large number of elements simultaneously) to the very nature of the simulated objects (fine-scale objects with complex shapes, interaction between rigid and deformable objects, as well as fluids, topology changes). If such large environments can be simulated adaptively, it would provide a great feeling of freedom to the user, who would be able to freely interact with a seemingly fine environment without being restrained by a small interaction space, or by a constraining interaction metaphor.

Should such a state be reached in the simulation and interaction performance be reached, it would create an great number of potential uses, from learning and training tools to endless recreation possibilities. This would also greatly increase the interest of the general public towards simulation and haptics in general, opening the way to even more everyday life uses.

5.2 Towards a better understanding of haptic perception

In this PhD, we showed that there was a psychophysical bias in haptic perception associated with the environment in which the interaction takes place. This bias might be linked with different interaction strategies in AR compared to VR. This shows that there is still a lot to investigate on haptic perception, and an ambitious objective might be to fully characterize the perception bias between AR and VR, with every aspect in the visual representation that is handled differently in AR compared to VR, including lighting conditions, integration inside the environment, presence of numerous objects around the main object or texture of the object. Fully characterizing the bias, and the interaction strategies, might help producing AR assistance tools that would influence the interaction as little as possible. A training tool involving a misperception of the task due to the nature of the environment would indeed be counter-productive, should the task require fine haptic interaction. Since there is an effect of AR compared to VR on perception and interaction, it would also be interesting to investigate what difference there is with a real interaction, both for AR and VR.

The advances in simulation methods and computation power should in the coming decade see the democratization of haptic interaction, which represents the next step after the recent success of VR displays among the general public. This should go on par with an always wider use of haptics and physically-based simulation for training application, with always more accurate simulation and interaction.

Publications from the author

A

Journals

- Y. Gaffary*, **B. Le Gouis***, M. Marchal, F. Argelaguet, B. Arnaldi, A. Lécuyer. *AR Feels "Softer" than VR: Haptic Perception of Stiffness in Augmented versus Virtual Reality.* To appear in the IEEE Transactions on Visualization and Computer Graphics (TVCG) issue of the 16th IEEE International Symposium on Mixed and Augmented Reality (ISMAR). 2017. (* both authors contributed equally to this work.)

International conferences

- **B. Le Gouis**, M. Marchal, A. Lécuyer, B. Arnaldi. *Elasticity-Based Clustering for Haptic Interaction with Heterogeneous Deformable Objects.* Proceedings of the 13th Eurographics Workshop on Virtual Reality Interaction and Physical Simulation. pp 75–83. 2017. *Honorable mention award*

- **B. Le Gouis**, F. Lehericey, M. Marchal, B. Arnaldi, V. Gouranton, A. Lécuyer. *Haptic Rendering of FEM-based Tearing Simulation using Clusterized Collision Detection.* Proceedings of IEEE World Haptics Conference. pp 406–411. 2017.

National conferences

- **B. Le Gouis***, Y. Gaffary*, M. Marchal, F. Argelaguet, B. Arnaldi, A. Lécuyer. *AR Feels "Softer" than VR: Haptic Perception of Stiffness in Augmented versus Virtual Reality.* To be presented at Journées de l'Association Française de Réalité Virtuelle. 2017. (* both authors contributed equally to this work.)

Other publication

- M. Achibet, **B. Le Gouis**, M. Marchal, P-A. Léziart, F. Argelaguet, A. Girard, A. Lécuyer, H. Kajimoto. *FlexiFingers: Multi-Finger Interaction in VR Combining Passive Haptics and Pseudo-Haptics.* Proceedings of IEEE Symposium on 3D User Interfaces. pp 103–106. 2017.

Appendix: Résumé long en français B

1 Introduction

Dans ce manuscrit de thèse, intitulé **Contribution à l'étude du rendu et de la perception haptique d'objets virtuels déformables**, nous présentons des travaux de recherches ayant pour objectif l'amélioration de l'interaction haptique avec des environnement virtuels utilisant de la simulation physique.

Avec le développement de dispositifs haptiques depuis début des années 1990, l'haptique est devenue une modalité cruciale pour interagir avec des environnements virtuels. Dérivée du grec *háptō*, "toucher", l'haptique définit ce qui est en lien avec le sens du toucher, et plus spécifiquement dans notre cas avec les interfaces utilisateur faisant appel au sens du toucher. D'abord développées comme un moyen de percevoir des données sans recourir à la vue, en affichant des surfaces [Minsky et al., 1990] ou des champs de forces volumiques [Brooks Jr et al., 1990], les interfaces haptiques permettent également d'interagir avec les environnements virtuels, offrant la capacité d'agir sur ceux-ci, et de ressentir les effets de cette action. L'haptique est une composante cruciale de nombreuses applications, notamment l'entraînement virtuel ou la téléopération. Dans le domaine de l'entraînement virtuel, notamment en entraînement de chirurgie où un chirurgien peut s'entraîner sur une opération spécifique sans avoir besoin d'un patient à opérer, l'interaction haptique doit être la plus similaire possible à l'interaction en conditions réelles.

Pour atteindre cette similarité, les objets virtuels doivent se comporter comme ils le feraient en réalité. Une méthode pour ce faire est la simulation physique. La simulation calcule la dynamique et le comportement des objets virtuels, comment ils répondent aux forces externes, aux contraintes internes, aux contacts, le tout en temps réel. La simulation d'objets déformables est notamment particulièrement intéressante du fait de leur omniprésence au quotidien, que ce soit dans les tissus vivants, les textiles ou encore dans la plupart de notre alimentation.

La plupart des applications interactives, notamment celles impliquant de l'interaction haptique, ont besoin d'allouer des ressources pour maintenir une certaine qualité (par exemple le taux de rafraîchissement) pendant la simulation, et particulièrement pour maintenir l'interactivité. Cela signifie qu'il faut faire un compromis entre la performance et la qualité du comportement de la simulation, car un comportement de meilleure qualité requiert plus de ressources, menant potentiellement à une perte d'efficacité qui pourrait nuire à l'interaction. Les méthodes adaptives, qui allouent les ressources dynamiquement pendant la simulation pour les attribuer là où elles sont le plus requises, offrent plus de flexibilité dans ce compromis. L'idée

sous-jacente de cette thèse est que la perception de l'utilisateur est un facteur clé pour déterminer comment la simulation doit être simplifiée, surtout si le comportement de l'objet est complexe. En effet, étant donné que l'interaction est accomplie par des êtres humains, il est crucial pour la fidélité à l'interaction réelle de prendre en compte la perception de l'utilisateur, être sûr qu'il/elle perçoit bien l'interaction comme elle se déroule. Si une partie de la simulation peut être simplifiée sans que l'utilisateur ne le remarque, il y a une amélioration globale de performances, sans perte dans la qualité perçue de la simulation, ce qui permet un compromis optimal.

Plus spécifiquement, il existe un certain nombre de défis inhérents à la création d'applications permettant l'interaction avec des objets complexes, et ce dans tous les aspects de la boucle d'interaction, à savoir la simulation physique des comportements complexes des objets, le rendu haptique de l'interaction avec ces objets, et enfin la perception humaine de cette interaction, et ce particulèrement avec l'émergence de diverses technologies d'affichage d'objets virtuels, qui pourraient l'influencer.

Simulation physique de comportements complexes pour les objets déformables

Pour créer une application permettant d'interagir avec des objets au comportement complexe, il faut commencer par la simulation de ces objets et comportements. L'un des premiers aspects à prendre en compte est que – en RV – une application utilisant de la simulation physique se doit d'être interactive, ce qui signifie que les méthodes de simulation doivent être suffisamment efficaces pour garantir cette interactivité. Déjà essentiel pour l'interaction avec des objets simples, ce besoin d'efficacité devient crucial lorsque l'objet a un comportement complexe, qui nécessite en général des performances en calcul supplémentaires.

La complexité d'un objet peut provenir de plusieurs apects. Dans un premier temps, la modalité sensorielle haptique permet de sentir des propriétés internes de l'objet qui peuvent ne pas être visibles depuis la surface. L'hétérogénéité de l'objet fait partie de ces propriétés. Étant donné que la plupart des objets de notre quotidien sont hétérogènes, il est donc nécessaire d'être capables de pouvoir les simuler, ce qui représente un premier défi. Un autre facteur de complexité important dans le comportement de l'objet est que lors de l'interaction avec un objet déformable, celui-ci peut subir un changement topologique. Les changements topologiques naturellement associés aux objets déformables sont la découpe et la déchirure. La simulation de découpe a reçu une attention importante de la part de la communauté scientifique, étant donné les nombreuses applications en simulation de chirurgie. La simulation de déchirure en revanche a été significativement moins étudiée, particulièrement en lien avec l'interaction haptique, et présente des défis uniques.

Interaction haptique avec des comportements complexes d'objets

Les interfaces haptiques ont des contraintes spécifiques qui doivent être prises en compte lors de l'élaboration d'applications interactives. Par exemple, le taux de rafraîchissement nécessaire lors de l'interaction avec des objets rigides pour avoir de bonnes sensations haptiques est de 1KHz [Colgate et al., 1995], bien plus que les 30 à

200Hz nécessaires à un bon rendu visuel. Cet ordre de grandeur de différence signifie que les méthodes ayant besoin d'interaction haptique doivent soit être un ordre de grandeur plus efficaces, soit trouver une manière de contourner le problème. Dans tous les cas, l'efficacité requise pour l'interaction haptique est un défi majeur, et tout particulièrement lorsque l'objet a un comportement complexe, qui utilise des ressources de calcul importantes.

En parallèle, avec des applications potentielles en entraînement de chirurgie, avec des chirurgiens exécutant des simulations de chirurgie en utilisant des interfaces haptiques, il est nécessaire d'avoir un rendu fidèle au comportement de l'objet. Pour que l'entraînement aie du sens, il faut que le comportement haptique soit le même (où à défaut le plus similaire possible) que dans les conditions réelles. Ceci vient en partie de la qualité de la simulation, mais aussi de la méthode de couplage haptique utilisée. Lors de l'interaction avec des objets au comportement complexe, cela signifie que la simulation de ce comportement complexe doit s'accompagner d'une méthode de rendu appropriée, ce qui représente un défi supplémentaire.

Influence de l'environnement sur la perception haptique

La perception haptique a reçu une grande attention en Réalité Virtuelle (RV), avec de nombreux résultats sur la perception de raideur, ou avec la prédominance du visuel sur la modalité haptique. L'haptique a également de nombreuses applications en Réalité Augmentée (RA), bien que moins répandues qu'en RV, avec notamment certains outils d'entraînement médicaux [Granados et al., 2017]. La plupart de ces applications utilisent la prédominance de la vision sur l'haptique pour suggérer des propriétés physiques sur les objets, que ce soit la sensation de rigidité de l'objet [Hirano et al., 2011] ou la sensation de poids [Hashiguchi et al., 2014]. De plus, il a été montré que certains biais de perception connus en RV sont aussi valables en RA, avec une différence potentielle d'intensité. Par exemple, le phénomène de sous-estimation des distances, bien documenté en RV, a aussi été constaté en RA, mais avec une importance moindre, ce qui suggère une influence de l'environnement (augmenté ou virtuel) sur la perception. Le fait que ce biais de perception existe en RA suggère une similarité dans la perception des objets virtuels dans les deux environnements. Cependant, le fait que l'intensité du phénomène soit différente suggère une influence de la nature de l'environnement, virtuel ou augmenté, sur la perception. La perception haptique en RA reste donc un champ de recherche ouvert, notamment en ce qui concerne la comparaison avec la RV.

Nos trois contributions consistent donc en :

1. Un modèle pour la simulation physique de matériaux complexes, et plus précisément la simulation physique d'objets hétérogènes multi-résolution;

2. Une méthode pour la simulation physique de déchirure de matériaux surfaciques déformables, permettant une interaction haptique complexe avec des objets subissant un changement topologique;

3. Une étude comparative sur la perception de raideur en fonction de deux technologies d'affichage, la RA et la RV.

2 Modèle de partition et d'aggrégation de l'élasticité pour la simulation multi-échelle d'objets hétérogènes

Introduction

Dans notre quotidien, nous sommes constamment confrontés à des matériaux hétérogènes, que ce soit dans le domaine médical, avec la plupart des organes du corps humain, ou dans la plupart de notre alimentation. Il est donc crucial de pouvoir les simuler de manière efficace pour permettre l'interaction, notamment haptique, avec des applications dans le domaine médical ou l'entraînement virtuel. Des méthodes à base de modèles embarqués [Nesme et al., 2009] ou des méthodes *frame-based* [Gilles et al., 2013] ont notamment été proposées pour simuler l'hétérogénéité des objets.

Une autre méthode pour simuler des objets hétérogènes de manière efficace est d'inscrire l'hétérogénéité dans la matrice d'élasticité de l'objet, pour une simulation utilisant la méthode des éléments finis (*MEF*), et d'utiliser les méthodes classiques d'amélioration de performances pour les objets déformables, à savoir de réduire la taille de la représentation géométrique de l'objet en rendant son maillage plus grossier [Imai et al., 2013]. L'un des défis associés à cette méthode est de déterminer quelle élasticité attribuer au maillage grossier, donc de savoir comment l'hétérogénéité est transmise au travers des différentes représentations de l'objet. En général, uniquement attribuer une élasticité unique moyenne ne convient pas, notamment lorsque l'objet est constitué d'élasticités très différentes [Imai et al., 2013]. De nombreuses méthodes ont été proposées pour attribuer des valeurs d'élasticité à un maillage plus grossier, notamment en utilisant de la dynamique inverse sur les deux résolutions du maillage [Kharevych et al., 2009] [Imai et al., 2013] [Chen et al., 2015], ou en embarquant l'élasticité dans la matrice d'élasticité de l'objet [Tagawa et al., 2013]. Ces différentes méthodes ne prennent toutefois pas en compte l'impact de ces modifications sur la perception haptique que l'on peut avoir lors de l'interaction avec ces objets.

Notre contribution est donc constituée de deux parties principales :

1. Une méthode d'attribution de l'élasticité pour un objet ayant une géométrie simplifiée, reposant sur une partition et une aggrégation des différentes élasticités dans l'objet initial (voir Figure B.1). Cette méthode prend en compte la répartition initiale de l'élasticité dans l'objet et modifie les valeurs d'élasticité dans l'objet pour rendre le processus d'attribution de l'élasticité plus simple à basse résolution, d'une manière qui affecte le moins possible la perception haptique lors de l'interaction avec cet objet;

2. Une évaluation de l'impact de la réduction de la résolution de l'objet sur la perception haptique.

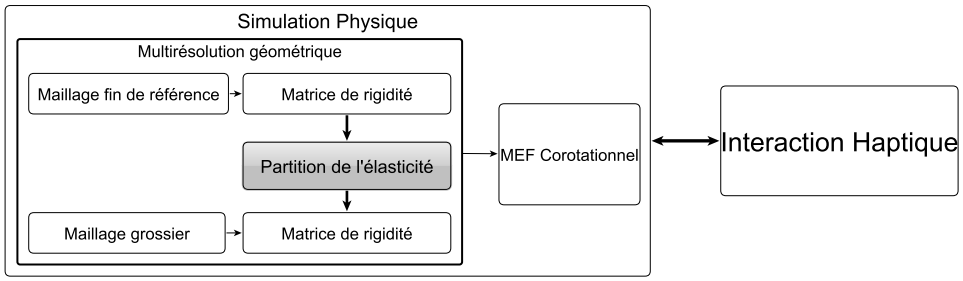


Figure B.1 – La partition de l'élasticité facilité l'attribution de l'élasticité à basse résolution pour le maillage grossier de l'objet.

Partition et aggrégation de l'élasticité

Le but de cette méthode est de faciliter l'attribution de l'élasticité pour les objets à basse résolution. L'idée sous-jacente à cette méthode est que l'attribution d'élasticité pour un maillage grossier se fait facilement si l'objet est homogène, ou constitué de larges volumes homogènes. Notre méthode crée donc de larges volumes homogènes en regroupant les tétraèdres d'élasticité similaire en prenant en compte leur répartition initiale dans l'objet. Le processus complet est illustré en Figure B.2.

Il comprend 4 grandes phases :

1. **L'initialisation** : Lors de cette phase sont définies dans un premier temps les élasticités qui seront considérées similaires par la suite, en les regroupant dans des classes d'élasticités. Pour ce faire, l'intervalle des valeurs d'élasticités est divisé en un nombre prédéfini d'intervalles de taille égales, qui deviendront nos classes. Ensuite, chaque classe d'élasticité est partitionnée au moyen d'un algorithme de k-moyennes. À ce stade, l'objet n'est pas modifié, pour chaque tétraèdre est uniquement ajoutée l'information de la classe et de la partition dans lequel il se trouve.

2. **L'homogénéisation** : Lors de cette phase, les tétraèdres ayant une élasticité similaire (dans la même classe d'élasticité donc) se voient attribuer une élasticité moyenne. Cette étape permet de réduire la complexité de l'objet en réduisant fortement le nombre d'élasticités différentes au sein de l'objet. De plus, les élasticités de chaque classe étant similaires, attribuer une valeure moyenne affecte peu le comportement de l'objet;

3. **L'aggrégation** : Lors de cette phase, les élasticités des tétraèdres sont changées pour regrouper ensemble les tétraèdres de même élasticité pour chaque partition, dans chaque classe, autour du barycentre des partitions, ce qui a pour effet de créer de larges volumes homogènes dans l'objet;

4. **Le grossissement géométrique** : Lors de cette étape, la répartition simplifiée de l'élasticité est utilisée pour attribuer aux tétraèdres du maillage grossier une

valeur d'élasticité. À l'intérieur d'un large volume homogène, l'unique valeur d'élasticité présente à haute résolution est simplement attribuée aux tétraèdres à basse résolution. Aux frontières de ces volumes, l'élasticité majoritaire à haute résolution détermine la valeur d'élasticité à basse résolution.

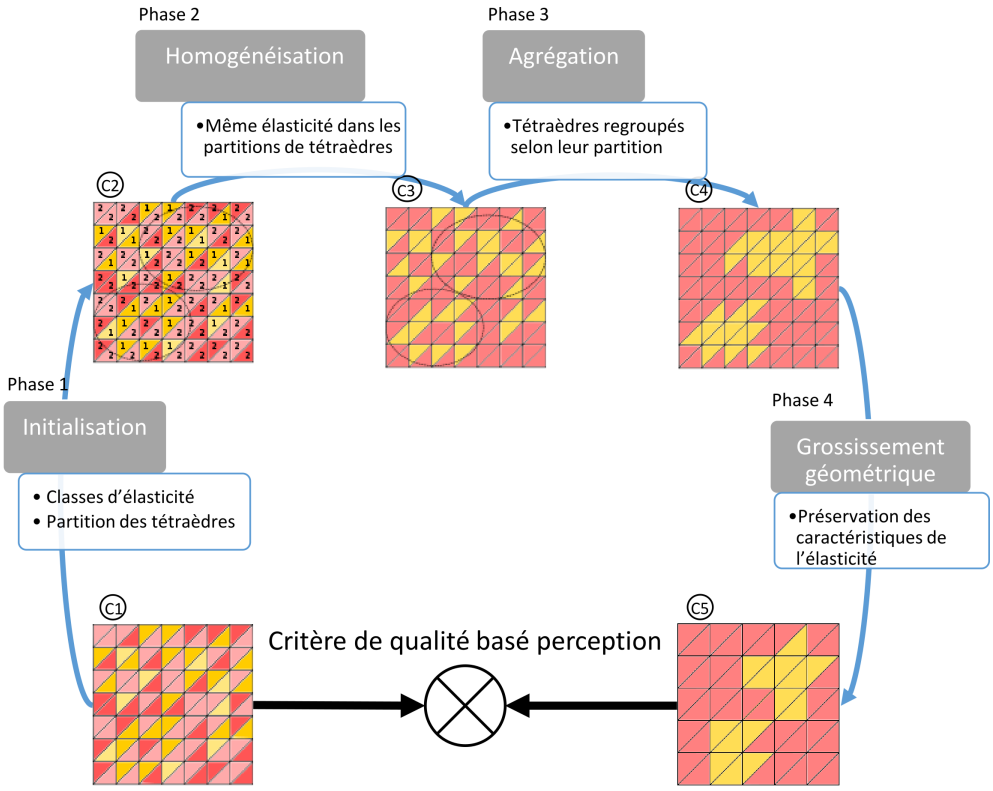


Figure B.2 – Processus d'attribution de l'élasticité pour les objets à basse résolution géométrique.

Évaluation

Une évalutation a été réalisée pour évaluer l'impact que la simplification a sur la perception de l'objet. Pour ce faire, nous comparons les forces lors de l'interaction avec l'objet initial et simplifié, avec un certain nombre d'objets de références. Nos objets consistent en un ensemble de 9 cubes, avec des répartitions initiales d'élasticité différentes. Les cubes sont compressés, et les forces correspondantes sont comparées. Pour que les forces entre l'interaction avec l'objet à haute et à basse résolution soient considérées similaires, nous utilisons un seuil de perception appelé seuil de discrimination. La loi de perception sous-jacente stipule que si l'écart relatif entre deux stimuli est suffisamment faible, les deux stimuli sont perçus comme étant égaux. Concernant la perception de force, ce seuil est de 10% [Jones and Hunter, 1990]. Pour l'ensemble de nos configurations d'élasticité dans nos cubes, nous comparons donc les forces lors de l'interaction avec l'objet initial avec celles lors de l'interaction avec l'objet basse résolution, et ce avec différentes valeurs pour le nombre de classes d'élasticités

et de partitions. Nous considérons que notre méthode fournit de bons résultats si l'on obtient des forces dont la différence relative est inférieure à ce seuil de discrimination de 10%. Les cubes testés comportaient 6000 tétraèdes pour les cubes de référence, et 3072 et 1296 tétraèdres pour les cubes à résolution plus faible. Dans un premier temps, nous avons constaté qu'une fois les valeurs d'élasticité modifées la valeur de force ne changeait pas significativement lorsque l'on changeait la résolution du cube, ce qui valide l'approche de créer de larges volumes homogènes. Uniquement homogénéiser l'ensemble du cube avec une valeur moyenne d'élasticité ne produisait par ailleurs pas de résultats satisfaisants pour la plupart des scénarios. Enfin, nous avons constaté que pour tous nos scénarios à l'exception d'un seul (pour laquelle aucune configuration satisfaisante n'a pu être trouvé avec un nombre raisonnable de partitions), on pouvait atteindre de bons résultats avec un nombre assez faible de partition (et donc de volumes homogènes à l'intérieur de l'objet), à savoir 6 partitions, ce qui constitue une validation supplémentaire de notre méthode.

3 Simulation physique de déchirure de surfaces déformables

Introduction

Les matériaux fins déformables, comme le papier, les vêtements, les sacs plastiques ou la nourriture, font partie intégrante de notre quotidien. Lorsqu'ils sont manipulés, ces objets subissent souvent des modifications topologiques en se déchirant. La simulation en temps réel de ces phénomènes, importante pour de nombreuses applications telles que l'entraînement à la chirurgie, les jeux vidéos ou l'animation, requiert d'importantes ressources en puissance de calcul, ce qui limite leur présence dans la plupart des applications nécessitant de l'interaction.

Contrairement à la simulation de découpe, très utilisée pour effectuer des simulations dans le domaine chirurgical, la simulation de déchirure a été beaucoup moins explorée, notamment dans le domaine de l'interaction haptique. Pour atteindre les performances requises pour l'interaction haptique, quelques aspects de la simulation doivent être gérés de manière optimale, car ils représentent des goulots d'étranglement en terme de coûts en calculs, à savoir la détection de collisions, et la simulation physique de déchirure. Cette contribution propose donc dans un premier temps une méthode de détection de collision optimisée pour les objets surfaciques, et dans un second temps une méthode de simulation physique de déchirure, avec une gestion optimisée de la propagation de déchirure.

Détection de collisions

La détection de collision calcule pour chaque paire d'objets ceux qui sont en collision. Cette détection s'effectue en plusieurs étapes. Premièrement, un élagage sur les paires d'objets est effectué si les objets sont suffisamment éloignés, ce qui permet de casser dans la majorité des cas la complexité quadratique qu'engendrerait le test de toutes les paires d'objets. Si deux objets sont suffisamment proches, une phase intermédiaire permet de ne tester que les parties de l'objet qui sont suceptibles d'être en contact,

notamment au moyen de hiérarchies de volumes englobants. La dernière étape est de calculer précisément les contacts, en faisant les tests directement sur les primitives (triangles, sommets, ...).

Pour cette contribution, nous nous sommes concentrés sur la phase intermédiaire. Nous avons utilisé une méthode itérative, qui réutilise les résultats d'un calcul de détection de collision d'un pas de temps de simulation sur l'autre, en exploitant la cohérence temporelle des objets [Lehericey et al., 2015] [Lehericey, 2016]. Si le déplacement relatif entre deux objets est suffisamment faible, les résultats des précédents calculs de collision restent valides, ce qui réduit grandement les calculs nécessaires. Afin de pouvoir prendre en compte la déformation des objets, ceux-ci sont décomposés en clusters avec un algorithme de k-moyennes, et les calculs de déplacement relatif sont effectués pour chaque cluster. Cette méthode est particulièrement efficace pour des scènes impliquant des objets avec un fort déplacement absolu, mais contenant certains objets qui ne se déplacent que très peu l'un par rapport à l'autre, comme par exemple un vêtement qui serait porté par un personnage.

Simulation physique de déchirure

La méthode de simulation de déchirure dépend intrinsèquement du modèle physique choisi pour l'objet. Par exemple, un objet représenté avec un système masses-ressorts est assez simple à déchirer, en coupant les ressorts qui subissent un allongement trop important [Terzopoulos and Fleischer, 1988]. Une telle représentation est rapide à calculer, mais n'est pas très fidèle pour le comportement physique de l'objet. D'autres méthodes, pour les tissus, simulent les fils indépendamment [Cirio et al., 2017]. Avec cette méthode aussi, la déchirure s'opère simplement, lorsque la contrainte appliquée au fil est trop importante. En revanche, cette méthode ne permet pas de simuler un tissu en temps réel et est donc incompatible avec une interaction haptique.

Pour cette contribution, nous proposons de simuler les objets avec la méthode des éléments finis corotationnelle. Avec cette méthode, la déchirure se crée à un endroit du maillage de l'objet pour lequel les contraintes dépassent un certain seuil [Gingold et al., 2004] [Allard et al., 2009], orthogonale à la direction principale de contrainte. Le calcul de cette direction principale étant coûteux, nous proposons de ne le faire qu'au voisinage de l'extrémité de la déchirure, une fois que celle-ci a commencé, améliorant ainsi les performances lors d'une propagation de déchirure. Nous utilisons également une méthode de *snapping* [Wu et al., 2014], qui déplace légèrement les sommets du maillage pour les mettre sur le trajet de la déchirure, ce qui réduit grandement le nombre de triangles créés lors de la déchirure, et permet donc de limiter la dégradation des performances lors de l'interaction.

Une illustration des résultats est présentée en Figure B.3, avec un mannequin portant un vêtement qui se fait déchirer. Le mannequin étant en mouvement, la détection de collision est près de 4 fois plus rapide qu'avec des méthodes standards, et la simulation de déchirure optimisée permet de gagner environ 2ms de calcul par pas de temps de simulation.

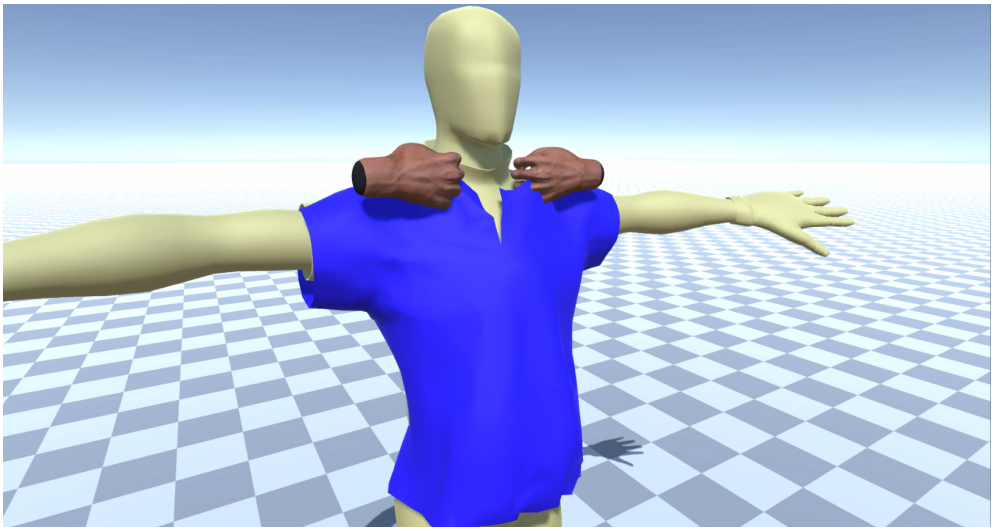


Figure B.3 – Exemple d'un mannequin se déplaçant, portant un vêtement qui subit une déchirure.

4 Étude comparative sur la perception de raideur entre des environnements augmentés et virtuels

Introduction

Comment interagit-on en Réalité Augmentée (RA)? Avec l'émergence des appareils de RA grand public et l'utilisation croissante d'applications de RA dans le domaine médical, allant des outils d'entraînement [Granados et al., 2017] à l'assistance pour la chirurgie mini-invasive [Nicolau et al., 2011] [Haouchine et al., 2013], la question est d'importance. La Réalité Augmentée et la Réalité Virtuelle (RV), bien qu'utilisant des technologies similaires, diffèrent fortement par le retour visuel proposé. Ce même retour visuel influence fortement le retour haptique [Srinivasan et al., 1996]. L'une des différences majeures vient de la présence ou non d'objets réels dans le champ de vision de l'utilisateur, aux côtés d'éléments virtuels. En revanche, il n'est pas clair de quelle manière cette différence influe sur la perception haptique de l'utilisateur.

La perception visuelle en RA a été largement explorée dans la littérature, avec l'étude de nombreux aspects de la RA comme l'environnment, l'appareil utilisé pour afficher les objets virtuels ou même les biais de perception de l'utilisateur [Kruijff et al., 2010]. Certains biais visuels bien documentés en VR, comme la sous estimation des distances, ont aussi été observés en RA [Jones et al., 2008], mais avec une amplitude moindre. Cependant, bien qu'il existe des études sur la perception visuelle en RA et les différences avec la VR, il existe peu d'études sur les autres modalités sensorielles, et particulièrement la modalité haptique.

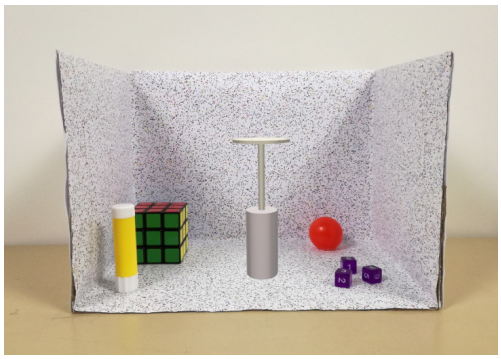
La perception haptique en RA est-elle différente de la perception haptique en RV? La présence d'objets réels en RA peut en effet altérer la manière d'interagir avec les objets virtuels, et ultimement la perception que l'on pourrait en avoir.

Dans cette contribution, nous avons conduit une expérience pour étudier à quel point la perception haptique de raideur est influencée lorsque l'environnement est augmenté ou virtuel.

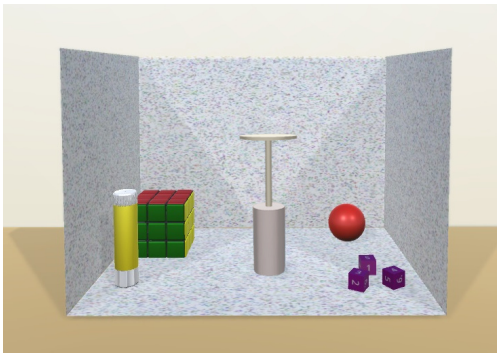
Expérience

Pour notre expérience, nous avons créé une scène composée d'éléments de la vie quotidienne, dans une boîte en carton, et une reproduction de cette scène dans un environnement virtuel. Pour l'expérience, les sujets devaient appuyer sur un piston virtuel qui était soit superposé à l'environnement réel (condition RA), ou intégré à l'environnement virtuel (condition RV). Les deux environnements sont montrés Figure B.4.

Pour afficher le piston et le cas échéant l'environnement virtuel, nous avons utilisé un casque de RA HoloLens, produit par Microsoft. Un masque de tissus était monté sur le casque, pour que tout le champ de vision effectif puisse être couvert par l'affichage augmenté. La condition RV consistait en un affichage de l'environnement virtuel sur l'intégralité du champ de vision du sujet, juste devant les objets réels pour permettre une bonne occlusion par l'HoloLens.



(a) Piston virtuel en RA.



(b) Piston virtuel en RV.

Figure B.4 – Conditions expérimentales. (a) Environnement RA avec un piston virtuel superposé dans une boîte en carton réelle. (b) Environnement RV avec le même piston virtuel situé dans la scène virtuelle.

Pendant l'expérience, les sujets devaient appuyer successivement sur le piston dans les deux conditions, avec différentes valeurs de raideur du piston, et pour chaque paire de piston RA/RV, déterminer lequel des deux pistons était le plus raide. L'ordre de présentation était contrebalancé pour éviter tout effet d'ordre. Le piston ayant la raideur de référence pouvait être soit le piston RA soit le piston RV.

Résultats

À partir des résultats de l'expérience, nous avons créé les courbes psychométriques correspondant à la condition où le piston RA était le piston de référence, et à la condition où le piston VR était le piston de référence. Les deux courbes, affichées Figure B.5, montrent une différence significative de 3.45% dans les seuils de perception

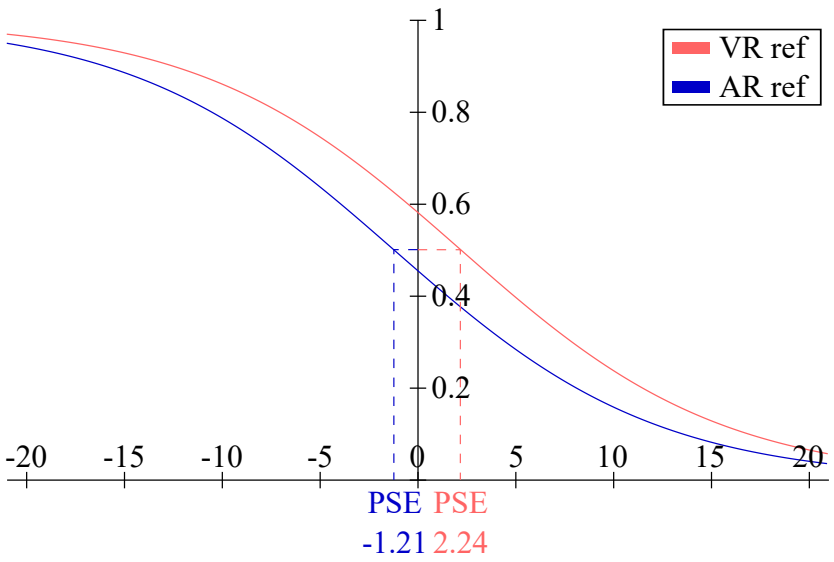


Figure B.5 – Courbes psychométriques RA et RV. La courbe rouge (resp. bleue) montre la courbe psychométrique avec la raideur de référence affichée dans l'environnement RV (resp. RA). Le Point Subjectif d'Équivalence (PSE) correspondant à chaque courbe est aussi indiqué. Les courbes ont été affichées grâce au logiciel dédié psignifit.

associés à chaque condition, ce qui indique que l'environnement a une influence sur la perception haptique des objets.

Nous avons également enregistré les forces appliquées par les sujets sur le piston, et la quantité de mouvement effectuée par les sujets lors de l'interaction avec chaque piston. Bien qu'ayant une quantité de mouvement similaire en RA et en RV, les sujets exerçaient des forces supérieures en RV par rapport à la RA, ce qui indique qu'ils appuyaient plus sur le piston RV que le piston RA. Il semblerait donc qu'il y ait une différence de stratégie lors de l'interaction avec un objet virtuel superposé à un environnement réel et lors de l'interaction avec un objet virtuel entouré d'un environnement virtuel. Une des explications possibles est que l'objet virtuel est plus cohérent avec son environnement en RV, et les sujets interagissaient avec cet objet avec plus de confiance que lorsque l'objet contrastait plus avec son environnement.

5 Conclusion

Cette thèse a pour but d'améliorer l'interaction haptique avec des environnements virtuels, et plus précisément avec des objets déformables. En particulier, trois défis nous intéressent particulièrement. Le premier défi est celui de la simulation physique de comportements complexes pour les objets déformables, et en particulier concernant les matériaux hétérogènes et les changements topologiques comme la déchirure. Le deuxième défi concerne l'interaction haptique avec ces objets au comportement complexe, avec des contraintes supplémentaires en terme d'efficacité et de fidélité du rendu. Le dernier défi est celui de l'impact que la technologie d'affichage peut avoir sur la perception haptique des propriétés d'un objet.

Dans un premier temps, nous avons présenté une méthode pour la simplification

géométrique d'objets hétérogènes. Cette méthode redistribue les valeurs d'élasticités à l'intérieur de l'objet pour créer de larges volumes homogènes, qui facilitent l'attribution des élasticités de l'objet à une résolution plus grossière. Cette redistribution prend en compte la répartition initiale des élasticités, afin d'avoir une élasticité à basse résolution la plus fidèle possible. Testée sur de nombreuses répartitions initiales, notre méthode fournit des résultats perceptuellement similaires pour la grande majorité d'entre eux, malgré un nombre de tétraèdres à basse résolution cinq fois inférieur à celui de l'objet initial.

Dans un second temps, nous avons présenté un pipeline pour la simulation physique de déchirure permettant l'interaction haptique. Afin d'atteindre les performances nécessaires à l'interaction haptique, nous avons traité deux aspects majeurs du pipeline qui représentent une part importante des besoins en ressources, la détection de collisions et la simulation de déchirure elle-même. La détection de collisions décompose les objets déformables en clusters pour pouvoir utiliser des méthodes incrémentales qui exploitent la cohérence temporelle sur les objets. La simulation de déchirure réduit les calculs nécessaires à la propagation de la déchirure, pour ne les faire qu'au voisinage de l'extrémité de la déchirure, évitant des calculs superflus.

Enfin, nous avons mené une étude utilisateur pour étudier l'impact de l'environnement (augmenté ou virtuel) sur la perception haptique de raideur. Pour ce faire, nous avons créé une scène réelle avec des objets de la vie quotidienne, que nous avons reproduite dans un environnement virtuel. Dans chacune des scènes était introduit un piston virtuel, avec lequel les participants devaient interagir. Les résultats montrent que le piston dans l'environnement virtuel était perçu comme plus raide que le piston virtuel augmenté dans la scène réelle, ce qui indique un biais sur la perception haptique en fonction de l'environnement. Nous avons également constaté que les participants interagissaient différemment avec le piston dans les deux environnements, ce qui pourrait suggérer une influence de l'environnement sur les stratégies d'interaction.

Appendix: Questionnaire subjectif C

Questionnaire subjectif

Merci d'avoir pris part à cette expérience. Nous allons maintenant vous poser quelques questions concernant l'expérience que vous venez de passer. Il n'y a pas de bonne ou de mauvaise réponse à ces questions, l'important est votre propre ressenti.

Pour chacune des affirmations suivantes, indiquez par une note allant de 1 à 7 votre degré d'accord avec cette affirmation en cochant l'une des sept cases.

1. Je suis familier avec la réalité augmentée.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

2. Je suis familier avec la réalité virtuelle.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

3. Je suis familier avec l'haptique.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

4. Le piston m'a paru réel en réalité augmentée.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

5. Le piston m'a paru réel dans la scène entièrement virtuelle.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

6. Je ne voyais pas l'environnement réel quand la scène était entièrement virtuelle.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

7. Hormis leur aspect réel/virtuel, je n'ai pas remarqué de différence entre les scènes (exemple : objet présent dans l'une mais pas dans l'autre).

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

8. Après l'expérience, je ressentais de la fatigue visuelle.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

9. Après l'expérience, je ressentais de la fatigue haptique.

1. Pas du tout d'accord | | | | | | | 7. Tout à fait d'accord

Pensez-vous que l'environnement réel a influencé votre perception haptique du piston virtuel ? Si oui, dans quel sens ?

Avez-vous des remarques générales sur cette expérience ?

List of Figures

List of Algorithms

List of Tables

Bibliography

[Allard et al., 2007] Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., Delingette, H., and Grisoni, L. (2007). Sofa-an open source framework for medical simulation. In *Proceedings of Medicine Meets Virtual Reality*, volume 125, pages 13–18. IOP Press. 51, 72

[Allard et al., 2009] Allard, J., Marchal, M., and Cotin, S. (2009). Fiber-based fracture model for simulating soft tissue tearing. *Studies in health technology and informatics*, 142:13–18. 69, 70, 108

[Avril et al., 2009] Avril, Q., Gouranton, V., and Arnaldi, B. (2009). New trends in collision detection performance. In *Proceedings of Virtual Reality International Conference*, volume 11, page 53. 23

[Ballesteros et al., 2005] Ballesteros, S., Reales, J. M., de León, L. P., and Garcia, B. (2005). The perception of ecological textures by touch: does the perceptual space change under bimodal visual and haptic exploration? In *Proceedings of the IEEE World Haptics Conference - First Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 635–638. 14

[Bandi and Thalmann, 1995] Bandi, S. and Thalmann, D. (1995). An adaptive spatial subdivision of the object space for fast collision detection of animated rigid bodies. *Computer Graphics Forum*, 14(3):259–270. 23

[Baraff and Witkin, 1998] Baraff, D. and Witkin, A. (1998). Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM. 69

[Barbič and James, 2008] Barbič, J. and James, D. L. (2008). Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics*, 1(1):39–52. 63

[Bender et al., 2014] Bender, J., Erleben, K., and Trinkle, J. (2014). Interactive simulation of rigid body dynamics in computer graphics. *Computer Graphics Forum*, 33(1):246–270. 18

[Botsch et al., 2007] Botsch, M., Pauly, M., Wicke, M., and Gross, M. (2007). Adaptive space deformations based on rigid cells. *Computer Graphics Forum*, 26(3):339–347. 28

[Bridson and Müller-Fischer, 2007] Bridson, R. and Müller-Fischer, M. (2007). *Fluid simulation: SIGGRAPH 2007 course notes*. ACM. 18

[Bro-Nielsen, 1998] Bro-Nielsen, M. (1998). Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490–503. 21

[Brooks Jr et al., 1990] Brooks Jr, F. P., Ouh-Young, M., Batter, J. J., and Jerome Kilpatrick, P. (1990). Project gropehaptic displays for scientific visualization. *ACM SIGGRAPH Computer Graphics*, 24(4):177–185. 1, 101

[Capell et al., 2002] Capell, S., Green, S., Curless, B., Duchamp, T., and Popovic, Z. (2002). Interactive skeleton-driven dynamic deformations. *ACM Transactions on Graphics*, 21:586–593. 29, 30, 117

[Chen et al., 2015] Chen, D., Levin, D. I., Sueda, S., and Matusik, W. (2015). Data-driven finite elements for geometry and material design. *ACM Transactions on Graphics (TOG)*, 34(4):74. 28, 104

[Chen, 2017] Chen, J. (2017). *Biomechanical Analysis of Different Aspects in Virtual Reality. Application*. PhD thesis, Université Grenoble Alpes. 91

[Chen et al., 1998] Chen, Y., Zhu, Q., Kaufman, A., and Muraki, S. (1998). Physically-based animation of volumetric objects. In *Proceedings of Computer Animation*, pages 154–160. 20

[Cholewiak et al., 2008] Cholewiak, S. A., Tan, H. Z., and Ebert, D. S. (2008). Haptic identification of stiffness and force magnitude. In *Proceedings of the IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 87–91. 12

[Chung, 1996] Chung, T. (1996). *Applied continuum mechanics*. Cambridge University Press. 21

[Cirio et al., 2014] Cirio, G., Lopez-Moreno, J., Miraut, D., and Otaduy, M. A. (2014). Yarn-level simulation of woven cloth. *ACM Transactions on Graphics*, 33(6):207. 68

[Cirio et al., 2017] Cirio, G., Lopez-Moreno, J., and Otaduy, M. A. (2017). Yarn-level cloth simulation with sliding persistent contacts. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1152–1162. 68, 69, 108

[Colgate et al., 1995] Colgate, J. E., Stanley, M. C., and Brown, J. M. (1995). Issues in the haptic display of tool use. In *Proceedings of the EEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 140–145. 3, 33, 102

[Congedo et al., 2006] Congedo, M., Lécuyer, A., and Gentaz, E. (2006). The influence of spatial delocation on perceptual integration of vision and touch. *Presence: Teleoperators and Virtual Environments*, 15(3):353–357. 81

[Courtecuisse et al., 2010] Courtecuisse, H., Jung, H., Allard, J., Duriez, C., Lee, D. Y., and Cotin, S. (2010). Gpu-based real-time soft tissue deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology*, 103(2):159–168. 22

[Davanne et al., 2002] Davanne, J., Meseure, P., and Chaillou, C. (2002). Stable haptic interaction in a dynamic virtual environment. In *Proceedings of the EEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2881–2886. 36

[Debunne et al., 2001] Debunne, G., Desbrun, M., Cani, M.-P., and Barr, A. H. (2001). Dynamic real-time deformations using space & time adaptative sampling. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36. 26, 27, 117

[Di Giacomo et al., 2007] Di Giacomo, T., Moccozet, L., Magnenat-Thalmann, N., Boulic, R., and Thalmann, D. (2007). Towards automatic character skeletonization and interactive skin deformation. In *Eurographics (STARs)*, pages 47–61. 31

[DiFranco et al., 1997] DiFranco, D. E., Beauregard, G. L., and Srinivasan, M. A. (1997). The effect of auditory cues on the haptic perception of stiffness in virtual environments. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume 61, pages 17–22. 16

[Drascic and Milgram, 1996] Drascic, D. and Milgram, P. (1996). Perceptual issues in augmented reality. In *Proceedings of Electronic Imaging: Science and Technology*, pages 123–134. International Society for Optics and Photonics. 13

[Duriez et al., 2006] Duriez, C., Dubois, F., Kheddar, A., and Andriot, C. (2006). Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):36–47. 24, 117

[Edwards and Bridson, 2014] Edwards, E. and Bridson, R. (2014). Detailed water with coarse grids: combining surface meshes and adaptive discontinuous galerkin. *ACM Transactions on Graphics*, 33(4):136. 31, 117

[Endo, 2016] Endo, H. (2016). Pressing movements and perceived force and displacement are influenced by object stiffness. *Physiology & behavior*, 163:203–210. 92

[Fierz et al., 2011] Fierz, B., Spillmann, J., and Harders, M. (2011). Element-wise mixed implicit-explicit integration for stable dynamic simulation of deformable objects. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 257–266. 25

[Fierz et al., 2012] Fierz, B., Spillmann, J., Hoyos, I. A., and Harders, M. (2012). Maintaining large time steps in explicit finite element simulations using shape matching. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):717–728. 25

[Frassinetti et al., 2002] Frassinetti, F., Bolognini, N., and Làdavas, E. (2002). Enhancement of visual perception by crossmodal visuo-auditory interaction. *Experimental Brain Research*, 147(3):332–343. 17

[Ganovelli et al., 2000] Ganovelli, F., Cignoni, P., Montani, C., and Scopigno, R. (2000). A multiresolution model for soft objects supporting interactive cuts and lacerations. *Computer Graphics Forum*, 19(3):271–281. 26, 27

[Garre and Otaduy, 2009] Garre, C. and Otaduy, M. A. (2009). Haptic rendering of complex deformations through handle-space force linearization. In *Proceedings of the IEEE World Haptics Conference - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 422–427. 38

[Gilles et al., 2013] Gilles, B., Faure, F., Bousquet, G., and Pai, D. K. (2013). Frame-based interactive simulation of complex deformable objects. In *Deformation Models*, pages 145–166. Springer. 30, 42, 104

[Gingold et al., 2004] Gingold, Y., Secord, A., Han, J. Y., Grinspun, E., and Zorin, D. (2004). A discrete model for inelastic deformation of thin shells. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 69, 108

[Glondu et al., 2010] Glondu, L., Marchal, M., and Dumont, G. (2010). A new coupling scheme for haptic rendering of rigid bodies interactions based on a haptic sub-world using a contact graph. In *Haptics: Generating and Perceiving Tangible Sensations*, pages 51–56. Springer. 37

[Granados et al., 2017] Granados, A., Perhac, J., Rosby, L. V., Lee, Y. M., Tan, G. W. L., Tan, T. C., Higham, J., Thalmann, N., Low-Beer, N., and Bello, F. (2017). See-through visualisation for training and assessing unsighted physical examinations. In *Proceedings of Eurographics 13th Workshop on Virtual Reality Interaction and Physical Simulation*, pages 085–092. 4, 77, 103, 109

[Greenish et al., 2002] Greenish, S., Hayward, V., Chial, V., Okamura, A., and Steffen, T. (2002). Measurement, analysis, and display of haptic signals during surgical cutting. *Presence: Teleoperators and Virtual Environments*, 11(6):626–651. 12

[Gregory et al., 2005] Gregory, A., Lin, M. C., Gottschalk, S., and Taylor, R. (2005). A framework for fast and accurate collision detection for haptic interaction. In *ACM SIGGRAPH Courses*, page 34. 63

[Gurari et al., 2009] Gurari, N., Kuchenbecker, K. J., and Okamura, A. M. (2009). Stiffness discrimination with visual and proprioceptive cues. In *Proceedings of the IEEE World Haptics Conference - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 121–126. 12

[Hahn, 1988] Hahn, J. K. (1988). Realistic animation of rigid bodies. *ACM SIGGRAPH Computer Graphics*, 22(4):299–308. 23

[Haouchine et al., 2013] Haouchine, N., Dequidt, J., Peterlik, I., Kerrien, E., Berger, M.-O., and Cotin, S. (2013). Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In *Proceedings of the*

IEEE International Symposium on Mixed and Augmented Reality, pages 199–208. 77, 109

[Hashiguchi et al., 2014] Hashiguchi, S., Sano, Y., Shibata, F., and Kimura, A. (2014). Rv dynamics illusion: Psychophysical influence on sense of weight by mixed-reality visual stimulation of moving objects. In *Proceedings of the International Conference on Virtual, Augmented and Mixed Reality*, pages 55–64. Springer. 4, 16, 103

[Hauth and Strasser, 2003] Hauth, M. and Strasser, W. (2003). *Corotational Simulation of Deformable Solids*. Citeseer. 22

[Heller, 1982] Heller, M. A. (1982). Visual and tactual texture perception: Intersensory cooperation. *Perception and psychophysics*, 31(4):339–344. 14

[Hinterseer et al., 2008] Hinterseer, P., Hirche, S., Chaudhuri, S., Steinbach, E., and Buss, M. (2008). Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems. *IEEE Transactions on Signal Processing*, 56(2):588–597. 13

[Hinterseer et al., 2006] Hinterseer, P., Steinbach, E., and Chaudhuri, S. (2006). Perception-based compression of haptic data streams using kalman filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 473 – 476. 13

[Hirano et al., 2011] Hirano, Y., Kimura, A., Shibata, F., and Tamura, H. (2011). Psychophysical influence of mixed-reality visual stimulation on sense of hardness. In *Proceedings of the IEEE International Conference on Virtual Reality*, pages 51–54. 4, 16, 103

[Hunter and Pullan, 2001] Hunter, P. and Pullan, A. (2001). Fem/bem notes. *Department of Engineering Science. The University of Auckland, New Zealand*. 21

[Hutchinson et al., 1996] Hutchinson, D., Preston, M., and Hewitt, T. (1996). Adaptive refinement for mass/spring simulations. In *Computer Animation and Simulation*, pages 31–45. Springer. 26, 27

[Imai et al., 2013] Imai, S., Yue, Y., Chen, B.-Y., and Nishita, T. (2013). Adaptively simulating inhomogeneous elastic deformation. In *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*, pages 237–244. 27, 28, 42, 104

[Jacobson et al., 2014] Jacobson, A., Deng, Z., Kavan, L., and Lewis, J. (2014). Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*. 31

[Jeon and Choi, 2008] Jeon, S. and Choi, S. (2008). Modulating real object stiffness for haptic augmented reality. In *Proceedings of the International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 609–618. 16

[Jones et al., 2008] Jones, J. A., Swan II, J. E., Singh, G., Kolstad, E., and Ellis, S. R. (2008). The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In *Proceedings of the ACM Symposium on Applied Perception in Graphics and Visualization*, pages 9–14. 13, 78, 109

[Jones and Hunter, 1990] Jones, L. A. and Hunter, I. W. (1990). A perceptual analysis of stiffness. *Experimental Brain Research*, 79(1):150–156. 12, 50, 106

[Jones and Tan, 2013] Jones, L. A. and Tan, H. Z. (2013). Application of psychophysical techniques to haptic research. *IEEE Transactions on Haptics*, 6(3):268–284. 13

[Jun et al., 2006] Jun, S., Choi, J., and Cho, M. (2006). Physics-based s-adaptive haptic simulation for deformable object. In *Proceedings of the IEEE 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 477–483. 27

[Kaldor et al., 2008] Kaldor, J. M., James, D. L., and Marschner, S. (2008). Simulating knitted cloth at the yarn level. *ACM Transactions on Graphics*, 27(3):65. 68

[Kaldor et al., 2010] Kaldor, J. M., James, D. L., and Marschner, S. (2010). Efficient yarn-based cloth with adaptive contact linearization. *ACM Transactions on Graphics*, 29(4):105. 68

[Kavan et al., 2010] Kavan, L., Sloan, P.-P., and O'Sullivan, C. (2010). Fast and efficient skinning of animated meshes. *Computer Graphics Forum*, 29(2):327–336. 30

[Kharevych et al., 2009] Kharevych, L., Mullen, P., Owhadi, H., and Desbrun, M. (2009). Numerical coarsening of inhomogeneous elastic materials. *ACM Transactions on Graphics*, 28(3):51. 28, 46, 104

[Klosowski et al., 1998] Klosowski, J. T., Held, M., Mitchell, J. S., Sowizral, H., and Zikan, K. (1998). Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36. 23

[Knapp, 2001] Knapp, J. M. (2001). *The visual perception of egocentric distance in virtual environments.* PhD thesis, ProQuest Information and Learning. 13

[Knorlein et al., 2009] Knorlein, B., Di Luca, M., and Harders, M. (2009). Influence of visual and haptic delays on stiffness perception in augmented reality. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 49–52. 14

[Kolcárek and Sochor, 2004] Kolcárek, P. and Sochor, J. (2004). Haptic rendering using velocity driven level of detail. In *Proceedings of Eurographics Workshop on Virtual Reality Interaction and Physical Simulation*, pages 149–158. 34, 117

[Kruijff et al., 2010] Kruijff, E., Swan, J. E., and Feiner, S. (2010). Perceptual issues in augmented reality revisited. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12. 13, 78, 109

[Kumazaki et al., 2007] Kumazaki, A., Terada, K., and Ito, A. (2007). Role of vision on haptic length perception. In *Proceedings of the IEEE World Haptics Conference - Second Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 336–341. 15

[Lécuyer, 2009] Lécuyer, A. (2009). Simulating haptic feedback using vision: A survey of research and applications of pseudo-haptic feedback. *Presence: Teleoperators and Virtual Environments*, 18(1):39–53. 17

[Lécuyer et al., 2000] Lécuyer, A., Coquillart, S., Kheddar, A., Richard, P., and Coiffet, P. (2000). Pseudo-haptic feedback: Can isometric input devices simulate force feedback? In *Proceedings of the IEEE International Conference on Virtual Reality*, pages 83–90. 17

[Lehericey, 2016] Lehericey, F. (2016). *Ray-traced collision detection : Quest for performance*. Theses, INSA de Rennes. 62, 64, 108

[Lehericey et al., 2013] Lehericey, F., Gouranton, V., and Arnaldi, B. (2013). New iterative ray-traced collision detection algorithm for gpu architectures. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 64

[Lehericey et al., 2015] Lehericey, F., Gouranton, V., and Arnaldi, B. (2015). Gpu ray-traced collision detection for cloth simulation. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 64, 108

[Lin and Otaduy, 2008] Lin, M. C. and Otaduy, M. A. (2008). *Haptic Rendering : Foundations, Algorithms, and Applications*. A K Peters, Ltd. 33, 56, 72

[Mahvash and Okamura, 2005] Mahvash, M. and Okamura, A. M. (2005). A fracture mechanics approach to haptic synthesis of tissue cutting with scissors. In *Proceedings of the IEEE World Haptics Conference - First Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 356–362. 38

[Manteaux et al., 2016] Manteaux, P.-L., Wojtan, C., Narain, R., Redon, S., Faure, F., and Cani, M.-P. (2016). Adaptive physically based models in computer graphics. *Computer Graphics Forum*. 25

[McGee et al., 2001] McGee, M. R., Gray, P., and Brewster, S. (2001). Feeling rough: multimodal perception of virtual roughness. In *Proceedings of Eurohaptics*, volume 2001, pages 29–33. 16

[McGee et al., 2002] McGee, M. R., Gray, P., and Brewster, S. (2002). Mixed feelings: Multimodal perception of virtual roughness. In *Proceedings of Eurohaptics*, volume 2002, pages 47–52. 16

[Mendoza and Laugier, 2000] Mendoza, C. A. and Laugier, C. (2000). A solution for the difference rate sampling between haptic devices and deformable virtual objects. In *Proceedings of International Symposium on Robotics and Automation*. 35

[Metaaphanon et al., 2009] Metaaphanon, N., Bando, Y., Chen, B.-Y., and Nishita, T. (2009). Simulation of tearing cloth with frayed edges. *Computer Graphics Forum*, 28(7):1837–1844. 69

[Minsky et al., 1990] Minsky, M., Ming, O.-y., Steele, O., Brooks Jr, F. P., and Behensky, M. (1990). Feeling and seeing: issues in force display. *ACM SIGGRAPH Computer Graphics*, 24(2):235–241. 1, 101

[Müller and Gross, 2004] Müller, M. and Gross, M. (2004). Interactive virtual materials. In *Proceedings of Graphics Interface*, pages 239–246. 22, 42, 51

[Müller et al., 2001] Müller, M., McMillan, L., Dorsey, J., and Jagnow, R. (2001). Real-time simulation of deformation and fracture of stiff materials. In *Proceedings of Computer Animation and Simulation*, pages 113–124. 71

[Narain et al., 2012] Narain, R., Samii, A., and O'Brien, J. F. (2012). Adaptive anisotropic remeshing for cloth simulation. *ACM Transactions on Graphics*. 69

[Nealen et al., 2006] Nealen, A., Müller, M., Keiser, R., Boxerman, E., and Carlson, M. (2006). Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4):809–836. 19, 69

[Nesme et al., 2009] Nesme, M., Kry, P. G., Jeřábková, L., and Faure, F. (2009). Preserving topology and elasticity for embedded deformable models. *ACM Transactions on Graphics*, 28(3):52. 28, 29, 41, 104, 117

[Nesme et al., 2005] Nesme, M., Payan, Y., and Faure, F. (2005). Efficient, physically plausible finite elements. In *Proceedings of Eurographics*. 22

[Nesme et al., 2006] Nesme, M., Payan, Y., and Faure, F. (2006). Animating shapes at arbitrary resolution with non-uniform stiffness. In *Proceedings of Virtual Reality Interaction and Physical Simulation*. 46

[Nicolau et al., 2011] Nicolau, S., Soler, L., Mutter, D., and Marescaux, J. (2011). Augmented reality in laparoscopic surgical oncology. *Surgical oncology*, 20(3):189–201. 77, 109

[O'brien et al., 2002] O'brien, J. F., Bargteil, A. W., and Hodgins, J. K. (2002). Graphical modeling and animation of ductile fracture. *ACM Transactions on Graphics*, 21(3):291–294. 69

[O'Brien and Hodgins, 1999] O'Brien, J. F. and Hodgins, J. K. (1999). Graphical modeling and animation of brittle fracture. In *Proceedings of the ACM 26th annual conference on Computer graphics and interactive techniques*, pages 137–146. 69

[Otaduy and Gross, 2007] Otaduy, M. A. and Gross, M. (2007). Transparent rendering of tool contact with compliant environments. In *Proceedings of the IEEE World Haptics Conference - Second Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 225–230. 38

[Otaduy and Lin, 2005a] Otaduy, M. A. and Lin, M. C. (2005a). Sensation preserving simplification for haptic rendering. In *Proceedings of ACM SIGGRAPH Courses*, page 72. 36, 37, 118

[Otaduy and Lin, 2005b] Otaduy, M. A. and Lin, M. C. (2005b). Stable and responsive six-degree-of-freedom haptic manipulation using implicit integration. In *Proceedings of the IEEE World Haptics Conference - First Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 247–256. 38

[Otaduy et al., 2009] Otaduy, M. A., Tamstorf, R., Steinemann, D., and Gross, M. (2009). Implicit contact handling for deformable objects. *Computer Graphics Forum*, 28(2):559–568. 24

[Overmars, 1992] Overmars, M. H. (1992). Point location in fat subdivisions. *Information Processing Letters*, 44(5):261–265. 22

[Palmer and Grimsdale, 1995] Palmer, I. J. and Grimsdale, R. L. (1995). Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116. 23

[Paloc et al., 2006] Paloc, C., Faraci, A., and Bello, F. (2006). Online remeshing for soft tissue simulation in surgical training. *IEEE Computer Graphics and Applications*, 26(6):24–34. 26, 27

[Payandeh et al., 2005] Payandeh, S., Dill, J., and Zhang, J. (2005). A study of level-of-detail in haptic rendering. *ACM Transactions on Applied Perception*, 2(1):15–34. 13, 35, 117

[Peterlik et al., 2011] Peterlik, I., Duriez, C., and Cotin, S. (2011). Asynchronous haptic simulation of contacting deformable objects with variable stiffness. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2608–2613. 36, 118

[Pfaff et al., 2014] Pfaff, T., Narain, R., de Joya, J. M., and OBrien, J. F. (2014). Adaptive tearing and cracking of thin sheets. *ACM Transactions on Graphics*, 33(4). 69

[Pressman et al., 2007] Pressman, A., Welty, L. J., Karniel, A., and Mussa-Ivaldi, F. A. (2007). Perception of delayed stiffness. *The International Journal of Robotics Research*, 26(11-12):1191–1203. 14

[Punpongsanon et al., 2015] Punpongsanon, P., Iwai, D., and Sato, K. (2015). Softar: Visually manipulating haptic softness perception in spatial augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1279–1288. 16

[Rolland et al., 1995] Rolland, J. P., Gibson, W., and Ariely, D. (1995). Towards quantifying depth and size perception in virtual environments. *Presence: Teleoperators and Virtual Environments*, 4(1):24–49. 13

[Ruspini et al., 1997] Ruspini, D. C., Kolarov, K., and Khatib, O. (1997). The haptic display of complex graphical environments. In *Proceedings of the ACM 24th annual conference on Computer graphics and interactive techniques*, pages 345–352. 33

[Sifakis et al., 2007] Sifakis, E., Shinar, T., Irving, G., and Fedkiw, R. (2007). Hybrid simulation of deformable solids. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 81–90. 28

[Souza et al., 2014] Souza, M. S., Wangenheim, A., and Comunello, E. (2014). Fast simulation of cloth tearing. *SBC Journal on Interactive Systems*, 5(1):44–48. 69

[Spence et al., 2004] Spence, C., Pavani, F., and Driver, J. (2004). Spatial constraints on visual-tactile cross-modal distractor congruency effects. *Cognitive, Affective, and Behavioral Neuroscience*, 4(2):148–169. 17

[Srinivasan and Basdogan, 1997] Srinivasan, M. A. and Basdogan, C. (1997). Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers and Graphics*, 21(4):393–404. 10, 11, 117

[Srinivasan et al., 1996] Srinivasan, M. A., Beauregard, G. L., and Brock, D. L. (1996). The impact of visual information on the haptic perception of stiffness in virtual environments. In *ASME Winter Annual Meeting*, volume 58, pages 555–559. 15, 78, 109

[Stewart and Trinkle, 2000] Stewart, D. and Trinkle, J. C. (2000). An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 162–169. 23

[Sulzer et al., 2007] Sulzer, J., Salamat, A., Chib, V., and Colgate, J. E. (2007). A behavioral adaptation approach to identifying visual dependence of haptic perception. In *Proceedings of the IEEE World Haptics Conference - Second Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 3–8. 16

[Susa et al., 2011] Susa, I., Sato, M., and Hasegawa, S. (2011). Multi-rate multi-range dynamic simulation for haptic interaction. In *Proceedings of the IEEE World Haptics Conference*, pages 233–238. 37, 38

[Swan et al., 2006] Swan, J., Livingston, M. A., Smallman, H. S., Brown, D., Baillot, Y., Gabbard, J. L., and Hix, D. (2006). A perceptual matching technique for depth judgments in optical, see-through augmented reality. In *Proceedings of the IEEE International Conference on Virtual Reality*, pages 19–26. 13

[Symmons and Richardson, 2009] Symmons, M. A. and Richardson, B. L. (2009). The equivalence of vision and haptics when matched spatiotemporally. In *Proceedings of the IEEE World Haptics Conference - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 582–586. 15

[Tagawa et al., 2013] Tagawa, K., Oishi, T., and Tanaka, H. T. (2013). Adaptive and embedded deformation model: An approach to haptic interaction with complex inhomogeneous elastic objects. In *Proceedings of the IEEE World Haptics Conference*, pages 169–174. 29, 104

[Tagawa et al., 2012] Tagawa, K., Yasuyuki, S., and Tanaka, H. T. (2012). Online remesh and multi-rate deformation simulation by gpu for haptic interaction with large scale elastic objects. In *Proceedings of the IEEE Haptics Symposium*, pages 533–540. 34

[Terzopoulos and Fleischer, 1988] Terzopoulos, D. and Fleischer, K. (1988). Modeling inelastic deformation: viscolelasticity, plasticity, fracture. *ACM SIGGRAPH Computer Graphics*, 22(4):269–278. 68, 69, 108

[Terzopoulos et al., 1987] Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. (1987). Elastically deformable models. *ACM SIGGRAPH Computer Graphics*, 21(4):205–214. 19

[Teschner et al., 2005] Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M.-P., Faure, F., Magnenat-Thalmann, N., Strasser, W., et al. (2005). Collision detection for deformable objects. *Computer Graphics Forum*, 24(1):61–81. 23

[Vroomen and Gelder, 2000] Vroomen, J. and Gelder, B. d. (2000). Sound enhances visual perception: cross-modal effects of auditory organization on vision. *Journal of experimental psychology: Human perception and performance*, 26(5):1583. 17

[Weller and Zachmann, 2009] Weller, R. and Zachmann, G. (2009). Inner sphere trees for proximity and penetration queries. In *Proceedings of Robotics: Science and Systems*, volume 2. 23

[Witkin et al., 1990] Witkin, A., Gleicher, M., and Welch, W. (1990). *Interactive dynamics*, volume 24. ACM. 23

[Wojtan et al., 2011] Wojtan, C., Müller-Fischer, M., and Brochu, T. (2011). Liquid simulation with mesh-based surface tracking. In *ACM SIGGRAPH 2011 Courses*, page 8. 18

[Wu et al., 2014] Wu, J., Westermann, R., and Dick, C. (2014). Physically-based simulation of cuts in deformable bodies: A survey. In *Proceedings of Eurographics (State of the Art Reports)*, pages 1–19. 69, 71, 108

[Wu et al., 1999] Wu, W.-C., Basdogan, C., and Srinivasan, M. A. (1999). Visual, haptic, and bimodal perception of size and stiffness in virtual environments. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume 67, pages 19–26. 17

[Yamamoto and Okamura, 2007] Yamamoto, T. and Okamura, A. M. (2007). Evaluation of human performance with kinematic and haptic errors. In *Proceedings of the IEEE World Haptics Conference - Second Joint EuroHaptics conference and*

Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pages 78–83. 15

[Zhang et al., 2002] Zhang, J., Payandeh, S., and Dill, J. (2002). Haptic subdivision: an approach to defining level-of-detail in haptic rendering. In *Proceedings of the 10th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 201–208. 13, 26, 27

[Zhang et al., 2003] Zhang, J., Payandeh, S., and Dill, J. (2003). Levels of detail in reducing cost of haptic rendering: a preliminary user study. In *Proceedings of the 11th IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 205–212. 13, 35

[Ziemer et al., 2009] Ziemer, C. J., Plumert, J. M., Cremer, J. F., and Kearney, J. K. (2009). Estimating distance in real and virtual environments: Does order make a difference? *Attention, Perception, and Psychophysics*, 71(5):1095–1106. 84

[Zilles and Salisbury, 1995] Zilles, C. B. and Salisbury, J. K. (1995). A constraint-based god-object method for haptic display. In *Proceedings of IEEE/RSJ International Conference on the Intelligent Robots and Systems 95*, volume 3, pages 146–151. 33

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Contribution à l'étude du rendu et de la perception haptique d'objets virtuels déformables

Nom Prénom de l'auteur : LE GOUIS BENOIT

Membres du jury :
- Monsieur LECUYER Anatole
- Monsieur ARNALDI BRUNO
- Madame MARCHAL Maud
- Madame COQUILLART Sabine
- Monsieur JAILLET Fabrice
- Monsieur DURIEZ Christian
- Monsieur ANDRIOT Claude

Président du jury : C. DURIEZ

Date de la soutenance : 21 Novembre 2017

Reproduction de la these soutenue

☑ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 21 Novembre 2017

Signature du président de jury

Le Directeur,

M'hamed DRISSI

Résumé

L'haptique joue un rôle majeur dans l'interaction avec des environnements virtuels, avec de nombreuses applications en entraînement virtuel, en prototypage et en assistance de téléopérations. En particulier, les objets déformables représentent un défi pour la simulation à cause de leur comportement intrinsèquement complexe. À cause des besoins particuliers en terme de puissance liés à l'interaction haptique, il est en général nécessaire de faire un compromis entre efficacité et précision, et tirer le meilleur parti de ce compromis reste un défi majeur. Les objectifs de ce doctorat sont premièrement d'améliorer l'interaction haptique avec des objets déformables au comportement complexe, et enfin d'étudier en quoi la perception peut nous aider dans cette tâche.

Dans cette thèse, nous proposons dans un premier temps un modèle pour la simulation physique d'objets hétérogènes déformables. Plus précisément, nous nous intéressons au problème de la multirésolution géométrique pour les objets hétérogènes, en nous concentrant sur la représentation de l'hétérogénéité à basse résolution des objets simulés. La contribution consiste en une méthode d'attribution de l'élasticité pour la basse résolution de l'objet, et une évaluation de ce changement de géométrie sur la perception haptique.

Nous nous intéressons ensuite à une autre classe de comportements complexes, les changements topologiques, en proposant un pipeline de simulation pour la déchirure haptique bimanuelle d'objets déformables fins. Cette contribution se concentre sur deux aspects essentiels à la simulation efficace de déchirure, à savoir la détection de collision pour les objets surfaciques, et la simulation physique de déchirure. La simulation est particulièrement optimisée pour la propagation de déchirure.

Le dernier aspect couvert dans cette thèse est l'influence de l'environnement sur la perception haptique de raideur, et plus particulièrement les environnements de Réalité Augmentée (RA). Comment perçoit-on les objets en RA par rapport à la Réalité Virtuelle (RV)? Est-ce que nous interagissons de la même manière dans ces deux environnements? Pour répondre à ces questions, nous avons mené une expérience pour comparer la perception haptique de raideur d'un piston virtuel entouré dans un premier cas d'objets de la vie quotidienne en RA, et du même piston entouré par une reproduction virtuelle de cet environnement réel en RV.

Ces contributions ouvrent de nouvelles perspectives pour l'interaction haptique avec des environnements virtuels, depuis la simulation efficace et fidèle d'objets déformables au comportement complexe à une meilleure compréhension de la perception haptique et des stratégies d'interaction.

Abstract

Haptics is a key part for the interaction with physically-based environments, with many applications in virtual training, prototyping and teleoperations assistance. In particular, deformable objects are challenging, due to the complexity of their behavior. Due to the specific need in performance associated to haptic interaction, a trade-off is usually necessary between accuracy and efficiency, and taking the best of this trade-off is a major challenge. The objectives of this PhD are to improve haptic rendering with physically-based deformable objects that exhibit complex behavior, and study how perception can be used to achieve this goal.

In this PhD, we first propose a model for the physically-based simulation of complex heterogeneous deformable objects. More specifically, we address the issue of geometric multiresolution for deformable heterogeneous objects, with a major focus on the heterogeneity representation at the coarse resolution of the simulated objects. The contribution consists in a method for elasticity attribution at coarser resolution of the object, and an evaluation of the geometric coarsening on the haptic perception.

We then focus on another class of complex objects behavior, topology changes, by proposing a simulation pipeline for bimanual haptic tearing of thin deformable surfaces. This contribution mainly focuses on two main aspects for an efficient simulation of tearing, namely collision detection for thin objects and efficient physically-based simulation of tearing phenomena. The simulation is especially optimized for tear propagation.

The last aspect that is covered by this PhD is the influence of the environment over haptic perception of stiffness, and more specifically of Augmented Reality (AR) environments. How are objects perceived in AR compared to Virtual Reality (VR)? Do we interact the same way on these two environments? In order to assess these questions, we conducted an experiment aiming at comparing the haptic stiffness perception of a piston surrounded by everyday life objects in AR and of the same piston surrounded by a virtual replica of the real environment in VR.

These contributions open new perspectives for haptic interaction with virtual environments, from the efficient yet faithful simulation of complex deformable objects behavior to a better understanding of haptic perception and interaction strategies.

INSA RENNES

N° d'ordre : 17ISAR 29 / D17 - 29
Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coësmes - CS 14315 - F-35043 Rennes Cedex
Tél : 02 23 23 82 00 – Fax : 02 23 23 83 96