FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Material changing haptics for VR

Henrique Melo Ribeiro

Mestrado em Engenharia Informática e Computação

Supervisor: Rui Pedro Amaral Rodrigues

Supervisor: Daniel Filipe Martins Tavares Mendes

July 26, 2023

Material changing haptics for VR

Henrique Melo Ribeiro

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Augusto Sousa
Referee: Prof. Rui Nóbrega
Referee: Prof. Rui Rodrigues

July 26, 2023

Abstract

Virtual reality is used in many contexts, whether in a professional setting or for personal use. However, in every single context, the main goal of VR is to immerse the user in its environment. Feedback is normally provided through the system's handheld controllers to increase the user's immersion. However, this feedback is most of the time minimal as they only provide vibrotactile feedback. There have only been a few improvements in haptic feedback for commercial usage, being mostly limited to vibrations for binary events. Some research has been done on material-changing haptics, more focused on grounded devices, however, research for commercial usage is limited.

There is interest in devices that are able to adapt to the virtual object being interacted with, changing the texture of the device's surface dynamically. On top of that, if the device is to be used in virtual reality, it is also interesting to use the device as a controller. As such, we will create a device that is simultaneously untethered and ungrounded, so that the user is able to move it freely as a virtual reality input device, while the device is able to dynamically adapt to the virtual environment.

By performing user tests, we aimed to validate the work done. The results obtained showed that the solution created has potential to mimic different types of materials, and that the same physical material has the ability to make the user perceive different virtual materials. We hope that the work done to obtain this solution can help in the development of future solutions focused on haptic feedback for VR.

Keywords:Virtual Reality, Haptics, Grabbable, Material Changing

Resumo

A realidade virtual é utilizada em muitos contextos, seja em ambiente profissional ou para uso pessoal. No entanto, em cada contexto, o principal objetivo da RV é a imersão do utilizador no seu ambiente. O feedback é fornecido por meio dos controladores portáteis do sistema para aumentar a imersão do utilizador. No entanto, esse feedback é mínimo, uma vez que estes sistemas apenas fornecem feedback tátil vibratório. Houve apenas algumas melhorias no feedback háptico para uso comercial, limitando-se maioritariamente a vibrações para eventos binários. Algumas pesquisas foram feitas sobre feedback háptico de mudança de material, mais focadas em dispositivos fixos, no entanto a pesquisa para uso comercial é limitada.

Um dispositivo de mudança de material deve ser capaz de se adaptar ao objeto virtual com o qual interage, alterando a textura da superfície do dispositivo dinamicamente. Além disso, se o dispositivo for usado em realidade virtual, também deve ser usado como controlador. Como tal, vamos criar um dispositivo que seja simultaneamente desconectado e não aterrado, para que o utilizador possa movê-lo livremente como um controlador de realidade virtual, enquanto o dispositivo é capaz de se adaptar dinamicamente ao ambiente virtual.

Através da realização de testes com utilizadores, procuramos validar o trabalho realizado nesta dissertação. Os resultados obtidos mostraram que a solução criada tem potencial para mimetizar diferentes tipos de materiais, e que um mesmo material físico tem a capacidade de fazer o utilizador sentir diferentes materiais virtuais. Esperamos que o trabalho realizado para obtenção desta solução possa auxiliar no desenvolvimento de soluções futuras focadas em feedback háptico para RV.

Palavras-chave: Realidade Virtual, Hápticos, Agarrar, Alteração de material

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Daniel Mendes and Rui Rodrigues, for all the patience and guidance they provided while helping me through the process and the opportunity to take part in such project. All the time spent making sure everything was going smoothly and the tips given throughout the development were essential for the project to reach its goal.

I'm extremely grateful to my family for listening and showing genuine interest when I talked about the state of the project, and for providing me invaluable emotional support when I was stressing out about any obstacle. A special thanks to my father, who helped me think of the materials to be used in the device, as well as search for ways to obtain them in the shortest time possible. To Diogo and Davide for making the environment at Porto the best it could be to develop this project, and for the extensive pilot testing.

I'd also like to thank every member of the GIG for welcoming me into the lab, and always having insightful feedback to help me face any challenges along the way. The light-hearted atmosphere inside the lab helped get through some long work days much better.

Finally, I'd like to recognize every single person that helped test the device at GIG. For taking time out of their day to help me finish this project, a huge thank you is due to these volunteers.

Henrique Ribeiro

"Research is seeing what everybody else has seen and thinking what nobody else has thought."

Albert Szent-Györgyi

Contents

1 Introduction **1**
- 1.1 Problem and Motivation . . . 1
- 1.2 Objective . . . 2
- 1.3 Document Structure . . . 2

2 Related Work **4**
- 2.1 Shape Changing Devices . . . 4
  - 2.1.1 Physical Alteration . . . 5
  - 2.1.2 Altered Shape . . . 5
- 2.2 Weight Mimicking Devices . . . 7
  - 2.2.1 Center of Mass Manipulation . . . 8
  - 2.2.2 Skin Manipulation . . . 9
- 2.3 Material Manipulation Devices . . . 10
  - 2.3.1 Texture Manipulation . . . 10
  - 2.3.2 Material Manipulation . . . 11
  - 2.3.3 Discussion . . . 13
- 2.4 Summary . . . 14

3 Creation of the Materializer **15**
- 3.1 Materializer: Designing The Material Changing Device . . . 15
  - 3.1.1 Requirements Specification . . . 15
  - 3.1.2 Concept . . . 16
  - 3.1.3 Needed Resources . . . 18
  - 3.1.4 Summary . . . 18
- 3.2 Materializer: Prototyping the Material Changing Device . . . 19
  - 3.2.1 Base Lid . . . 19
  - 3.2.2 Prisms . . . 20
  - 3.2.3 Reutilized Pieces . . . 23
  - 3.2.4 Final Device . . . 24
  - 3.2.5 Summary . . . 25
- 3.3 Materializer: Controlling the Device . . . 27
  - 3.3.1 Architecture . . . 27
  - 3.3.2 Microcontroller . . . 27
  - 3.3.3 Unity Application . . . 29
  - 3.3.4 Summary . . . 31

4 Evaluation **32**
4.1 Procedure . . . 32
4.2 Selected Materials . . . 33
4.3 Participants . . . 36
4.4 Tasks . . . 36
4.5 Setup . . . 38
4.6 Prototype . . . 38
4.7 Results . . . 39
4.7.1 Task 1 . . . 39
4.7.2 Task 2 . . . 40
4.7.3 User Form . . . 43
4.8 Summary . . . 45

5 Conclusions **46**
5.1 Future Work . . . 47

References **49**

A Materializer Version History **52**

B Test Forms **54**

C Microcontroller Code **56**

List of Figures

2.1 XRings showcase mimicking 2 different objects [10]. 5
2.2 Fully expanded and compressed *Shape-a-getti* [2]. 6
2.3 *PoCoPo* device mimicking three different objects [26]. 6
2.4 Grabbing a virtual object with *Grabity* [7]. 7
2.5 Perceived objects compared to a mold created by Fujinawa et al.'s perception model [8]. 7
2.6 *Shifty* and its inner components [27]. 8
2.7 *Transcalibur* configurations [17]. 8
2.8 Multiple applications for *ReCompFig* [25]. 9
2.9 Sakurada et al.'s device and the different stages of material recognition [16]. . . . 10
2.10 Germani et al.'s device [9] 11
2.11 *DualVib*'s two types of haptic feedback [21]. 12
2.12 Configuration of *HapticRevolver* [24]. 12
2.13 *Snake Charmer* with a cube attached [1]. 13

3.1 A simple example of a Trivision mechanism at work. 17
3.2 Initial design of the device using the *Trivision* mechanism. 17
3.3 Side by side view of Materializer and Shape-a-Getti. 19
3.4 Upgraded lid component with thicker connector component 20
3.5 Updated design of the prism used in the device. 21
3.6 Closer look at the gap where the material slides into. 21
3.7 Image of a prism used to hold the materials. 22
3.8 *LEGO Technic* axel at the bottom of the prism. 22
3.9 Empty base used to hold the hardware 23
3.10 Base filled with hardware 23
3.11 Main gear. 24
3.12 Small gear. 24
3.13 Base without the lid. 24
3.14 Roof used for the device. 25
3.15 Device next to water bottle. 26
3.16 Exploded view of the device. 26
3.17 Architecture of proposed solution 27
3.18 Microcontroller's process diagram. 28
3.19 Inspector view of the *SteamVR_TrackedObject* class. 30

4.1 Material strips. 35
4.2 Device with metallic surface. 35
4.3 Device with wooden surface. 35

4.4 Device with foam surface. 35
4.5 Example scenario of the first task. 36
4.6 Example scenario of the second task. 37
4.7 Physical environment used for the tests. 38
4.8 Virtual environment used for testing. 39
4.9 Confusion matrix for the first task. 40
4.10 Most popular combinations for the second task. 41
4.11 Percentages of guesses with only one material. 42
4.12 Responses to the first statement the users answered. 43
4.13 Responses to the second statement the users answered. 44
4.14 Number of materials the users thought the device had. 45

List of Tables

2.1 Characteristics of haptic devices 14

4.1 Characterization of different materials. 34
4.2 Order of simulated materials. 37
4.3 Table containing the percentage of answers with at least one correct answer and the respective confidence. 41
4.4 Table containing the percentage of answers with at least one incorrect answer and the respective confidence. 42
4.5 Interquartile range and median of the first two statements. 44

B.1 Post Test Statements 55

Abbreviations

VR	Virtual Reality
HMD	Head-Mounted Display

Chapter 1

Introduction

Virtual reality allows the users to immerse themselves in a different environment, in an attempt to make them feel like they are in it. This feeling of immersion can be amplified by using haptic devices that provide feedback to the users when interacting with the environment they are inserted into.

With the increase in the popularity of virtual reality, there has also been a considerable increase in applications created for it. The more applications there are, the more complex the interactions the users can make with the virtual objects become. In order to truly immerse the users into the environment, even the devices that connect them to said environment must feel natural. Currently, the most commonly used approach is to provide the user with vibrotactile feedback when interacting with the different objects, but more is needed. To create a more immersive experience, the controllers that serve as input should adapt to the objects at hand and change some characteristics, such as shape, weight, material, and texture.

1.1 Problem and Motivation

The user's experience is limited by the fact that different ways to provide feedback are not being explored for public access. Most of the efforts to develop virtual reality technology have been focused on visual and auditory feedback, leaving the physical aspect of the feedback with little to no research. Many studies have been conducted in order to create different devices capable of recreating multiple different aspects of the virtual object the user is interacting with, such as the shape and weight of an object. However, only some have tried to mimic the material and texture.

Some of these solutions can be found in section 2, but most of the solutions are either grounded or too limited to be used in a wide variety of applications. By having a re-configurable hand-held device, it is possible to create multiple configurations to adapt to the object the user is interacting with, without having the need to completely alter the haptic device.

By researching the possibilities that hand-held devices can provide to the user, it is possible to enrich the sentiment of "being there" and improve the user's interaction beyond simple vibrations.

1.2 Objective

The goal of this thesis is to create a more immersive way for the user to interact with the virtual objects around him, simulating materials that the user encounters in said environments. To achieve this simulation, an ungrounded hand-held haptic device, called *Materializer*, was created to be used in virtual reality. The solution must adapt to represent the virtual object being interacted with, changing its apparent material, and improving the computer-user interaction.

The device should be able to change the material of the surface the user holds in an acceptable time, not to hinder the user's experience. The device should also be adaptable enough to swap to new materials without the need of creating a new device.

With this in mind, the objective of this thesis can be summarized in the following topics:

- Design and develop a solution that allows for the simulation of multiple materials or textures;
- The solution should be generic so that adding new materials can be done so with little effort;
- The solution should serve as an input system to allow the user to interact with the virtual environment.

1.3 Document Structure

This document is composed of five different chapters. The first one is the chapter this section in included into. The first chapter's purpose is to describe the goal of the thesis, as well as the motivation behind it.

The second chapter focuses on related work. It explores what has already been done and summarizes what has already been concluded in other researches. It contains a section explaining the importance of the different approaches and why they work in their relative field. After that, a section is presented, taking a closer look at devices whose objective is to mimic an object's shape and/or weight, followed by another section that reports the related work found in the texture and material manipulation field. Finally, to close this chapter, a discussion, and comparison of all approaches are done.

The third chapter will explain the implemented solution. This chapter contains some insight into the brainstorming done to reach this proposal, as well as the improvements done to the early brainstorming. This chapter also contains the creation of the device. It includes representations and specifications for the pieces of the device that were printed using a 3D printer, as well as their functionality. Some problems with the structure are also highlighted in this chapter. Finally, this chapter explains how the device is controlled. It explains some functional design choices made for the device's software architecture. Additionally, this chapter explains the decisions made in both the *Unity* code and the code burnt into the microcontroller. Finally, this chapter explains how the messages are communicated between the microcontroller and the application and some examples of the configuration file used to configure the application with the correct rotation values.

The fourth chapter describes how the solution was tested and validated. This description contains information about the participants and how the testing environment was set up. This chapter informs how the tests with the users were conducted, and what information was extracted from these tests. Finally, the results are analyzed and conclusions are drawn from the analysis done.

The final chapter closes with the conclusions that can be taken from the work done in this project and document, and some possible future work.

Chapter 2

Related Work

The haptic feedback provided when interacting with virtual reality available to the public is limited. The most commonly used VR system at the date of writing is the *HTC Vive*[1], an ungrounded combination of devices composed of two handheld controllers and a virtual reality headset (with some additional tracking hardware). This set, although covering several interaction modalities, lacks when it comes to providing haptic feedback to the user. The feedback the controllers give consists of only vibratory feedback during events.

Much of the literature analyzed to kick-start our work, report innovations done on how the feedback is given to the user through multiple channels or simple hardware, mainly focusing on a single area. This chapter of the document will explore the different ways the feedback is provided and compare them at the end of the chapter.

Contrary to the common haptic feedback provided by the commercial VR system, the goal of this project is to develop a device that explores other means of providing haptic feedback. Instead of relying on the commonly used vibrotactile feedback, the device created should be able to change some of its characteristics. This allows the device to provide the users with more realistic haptic feedback.

In section 2.1 we will analyze the various devices used to mimic the shape of a virtual object. In section 2.2 we will analyze how the devices can manipulate their center of mass to produce a feeling that the weight of the said device is changing dynamically. Finally, in section 2.3, we will analyze the already created approaches to change a device's material and create some parallels on the proposed device.

We consider all these areas vital to the work, as some ideas that will be used to create our final product originated from these multiple areas.

2.1 Shape Changing Devices

We classify *Shape displays* devices as devices created in order to mimic multiple shapes used in different virtual environments, through the use of ungrounded controllers. If we are to recreate

[1] HTCVive (accessed on 25/Oct/2022): https://www.vive.com/us/

every possible configurations, the device would need to be very robust and costly, additionally, the device would need to have multiple degrees of freedom to accommodate all the possible shapes. To reduce the complexity of the devices, it is possible to provide the user with visual and auditory queues to influence the perception of the user in relation to the mimicked shape [8].

2.1.1 Physical Alteration

In XRings [10], as seen in Fig. 2.1, the device is composed of multiple rings, making it able to change their radius. By doing this, the device can shift every layer to best match the object being interacted with. These rings, however, can only represent symmetric shapes with a limited radius. The *XRings* device can also detect when the user is forcing the rings in and uses it as an input system for the controller.

Figure 2.1: XRings showcase mimicking 2 different objects [10].

Another approach to this paradigm is presented in *Shape-a-getti* [2]. This approach involves creating a device inspired in Chuck Hoberman's shape-changing sculptures, allowing the machine to adopt multiple radially symmetrical shapes using a single actuator, as can be seen in Fig. 2.2. This actuator is a motor in the controller's base that rotates a gear connecting every pole. To counter the force the user's grasp makes, a mechanically-activated brake is used.

PoCoPo [26] takes an entirely different approach to shape-changing devices. Instead of altering the shape of the device by rotating or expanding components, it instead uses pins that can be moved up and down. By doing this, it is possible to create different configurations of the device. Worm-gears are responsible for moving the pins, so they are not driven back by the users' strength. The main downside of this device, as written in [26], is the fact that the device can only create 2.5D objects. There are only two sides with pins, as seen in Fig. 2.3, leaving the other faces of the device with no feedback whatsoever.

2.1.2 Altered Shape

Instead of changing their shape, the devices in this section try to make the users think they did. In the case of *Grabity* [7], instead of changing its form, it instead uses brakes. When the users try

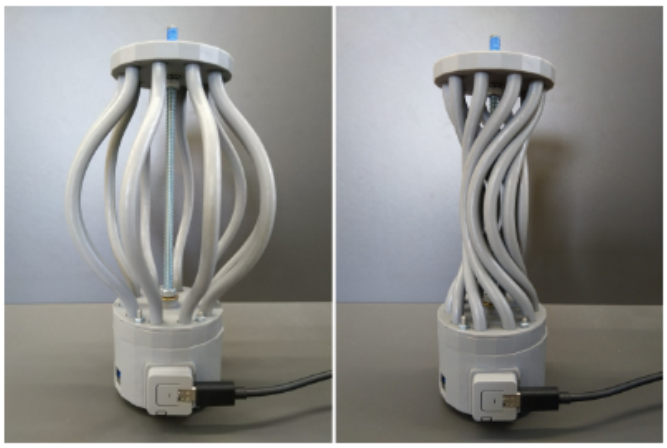

Figure 2.2: Fully expanded and compressed *Shape-a-getti* [2].

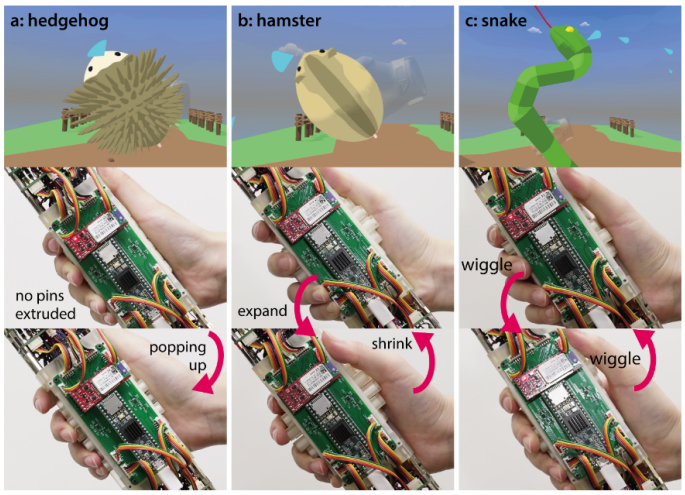

Figure 2.3: *PoCoPo* device mimicking three different objects [26].

to interact with the virtual objects, *Grabity* configures the breaks present in each sliding part to create a rigid feedback force once the users pinch the virtual object, which is represented in Fig. 2.4. In the second image, the brakes stop the movement as the device has reached the point where the users grab the object. This force feedback, paired with what is displayed to the users, makes for an immersive experience where the users genuinely believe they are grasping an object. The main downside is that the device is limited to flat objects, not handling curved surfaces well.

Eisuke Fujinawa et al. [8] propose a design method for hand-held VR controllers that create an illusion of an object's shape. With help from user input to indicate the maximum length and the shape of the virtual object, it is possible to create a mold to hold weights. These inputs are put through a perception model, which predicts the perceived shape for an arbitrary controller. With this design, it is possible to create small molds that resemble bigger objects by placing weights in gaps in said mold, as seen in Fig. 2.5. This resemblance is further accentuated by the objects the users interact with within the virtual world. The main problem of this approach is that one device cannot provide the users with multiple shapes, needing to create multiple to mimic various shapes

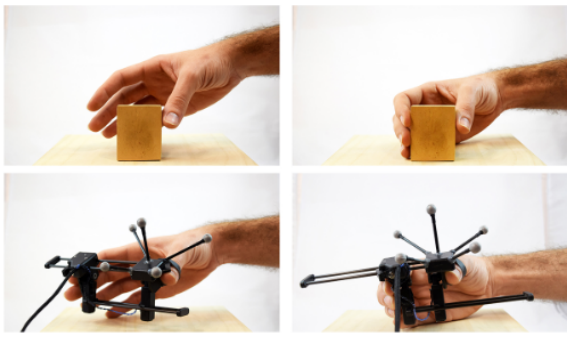

Figure 2.4: Grabbing a virtual object with *Grabity* [7].

or manually move the weights in the mold.

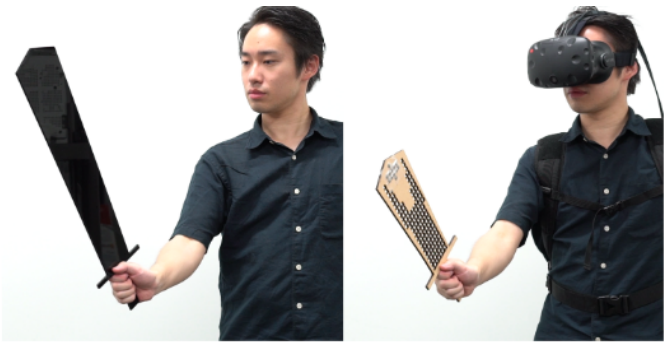

Figure 2.5: Perceived objects compared to a mold created by Fujinawa et al.'s perception model [8].

Other solutions explored for shape-changing devices are:

- *Haptobend* [14]: a device that allows the user to manually change its configuration;
- *Ninja Track* [13]: a device that uses a belt-like shape, that is able to change between a flexible mode and a rigid mode;
- *Elastilinks* [23]: a device that uses an elastic which moves around on the device, so that the point of application of force when pulling it changes, mimicking different objects.

These devices were studied and are present in the comparison between solutions in section 2.3.3.

2.2 Weight Mimicking Devices

When interacting with objects in the real world, the weight of the object is very important for the object's perception. With this in mind, many ways to recreate the feel of the object's center of mass have been explored. The most common way to mimic the objects with different centers of mass is through the use of moving weights, which, by being further from the handle where the user holds the device, makes it feel heavier [27]. Additionally, instead of having a weight move, causing

the center of mass to shift position, there are other ways to mimic the weight of an object. Some other ways are how the object interacts with the skin and how the user perceives inertia. These two methods will be explored further in this section, as well as the moving weight to change the device's center of mass.

2.2.1 Center of Mass Manipulation

The simplest approach studied was a device called *Shifty* [27]. The device is composed of a hollow plastic tube with a weight inside. This weight is then moved around inside the tube to make the device feel heavier or lighter than it is, depending on how far away the weight has been moved from the handle. The device and its inner components can be seen in Fig. 2.6

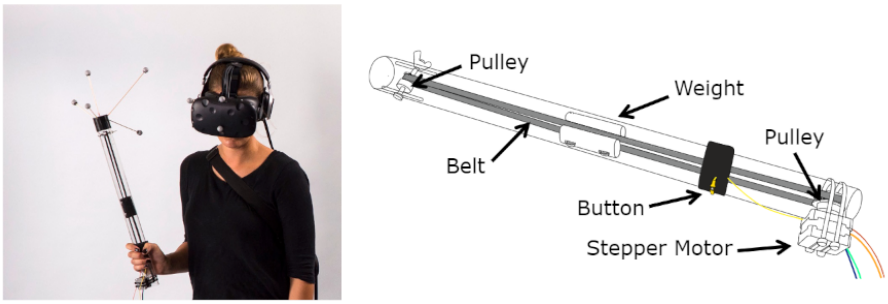

Figure 2.6: *Shifty* and its inner components [27].

A similar device to *Shifty* is called *Transcalibur* [17]. *Transcalibur* tries to fix a problem that *Shifty* has, which is the low degrees of freedom the device provides. By placing the weights on two arms that can change the angle they make with the handle, it is possible to adapt the center of mass to different positions, adding a new degree of freedom and making the number of objects that can be recreated larger. *Transcalibur*'s device can be seen in Fig. 2.7.

Figure 2.7: *Transcalibur* configurations [17].

ReCompFig [25] takes a different approach from the one just described. Instead of moving weights around to displace the center of mass of the device, it describes a framework that allows for the creation of devices that simulate the weight shift of a virtual object using weights connected by a cable to a handle. With the help of multiple tensioning cables, the same device can be used to simulate the weight of multiple objects. This is achieved by loosening or tightening different tensioning cables connected to the weight, increasing or limiting the number of degrees of freedom.

An example of a device using *ReCompFig*'s framework can be seen in Fig. 2.8. In these images, Recompfig is being used to simulate two situations. In mode one, every tensioning cable

is loose, allowing for multiple degrees of freedom, simulating a container with liquid, allowing the weight to move in every axis. Mode 2 has some tensioning cables tightened, creating fewer degrees of freedom, only allowing the weight to move in a single axis, creating a simulation of a flexible ruler.

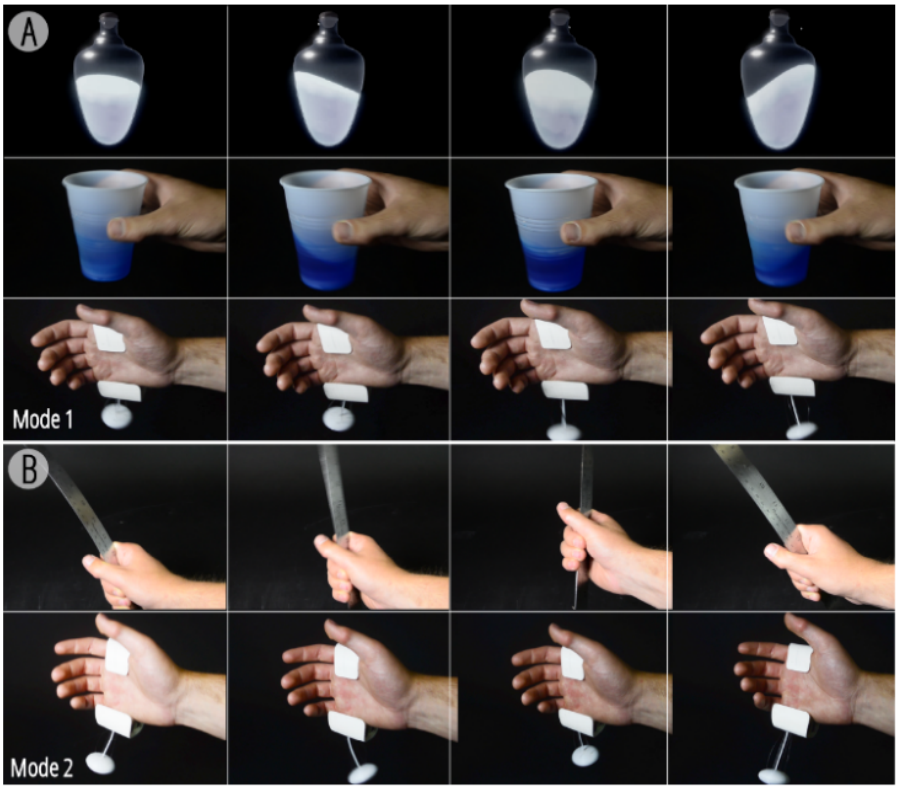

Figure 2.8: Multiple applications for *ReCompFig* [25].

2.2.2 Skin Manipulation

The skin receptors can recognize specific forces, textures, and many other stimuli. This particular characteristic is explored by Grabity, which gives the user an enhanced sense of weight.

On top of manipulating its apparent shape, Grabity also simulates the grasping of an object by using skin-stretching mechanisms, deforming the fingertip of the user as if they were holding the virtual object. This deformation is perceived directionally, akin to gravity, thus providing the user with directional cues where the force is being applied.

To truly simulate the object's weight, two actuator coils are used in each of the fingers holding the device. These actuators move up and down, one at a time, to create a downward or upward force. The users then perceive this force as gravity pulling down on the object.

The solution proposed by Tsai H. et al.[22] was also studied and was compared with every other studied solution in section 2.3.3. This solution proposed a device with a weight bound to rubber bands, that by controlling the elasticity of the rubber bands, the weight's movement is restrained or freed, functioning as the virtual object's oscillating center of mass.

2.3 Material Manipulation Devices

The last object characteristic our research focused on was the object's material and texture. Holding an object is very different, even if the weight and shape are the same, if its consistency or texture is different. Because of this, it is essential also to study this area. In the research, it was noticed that most devices to simulate texture are grounded, forcing the user to stay close to support the device, limiting the movement.

It was also noticed that there does not seem to be an agreed-upon definition for material and texture. Because of this, we based the definition for these terms on the Merriam Webster's[2] dictionary definition. Here, *texture* is defined as a visual or tactile characteristic of a surface, for example the texture of an oil painting. A *material* is defined as a matter with qualities that give it individuality and can be categorized, for example, a material can be sticky or slippery.

2.3.1 Texture Manipulation

The approach proposed by Sakurada et al. [16] consists of a sensor connected to a device containing multiple materials with different textures. The sensor should be placed on top of the user's finger and should be able to capture the material being interacted with. The sensor's readings are then processed with feature extraction methods to classify the materials being touched. After this, the device picks the correct material, pushes it against the user's skin, and rotates it, depending on the accelerometer reading of the sensor. This mechanism, shown in Fig. 2.9, provides the user feedback on the texture being touched in a location other than the fingertip. This behavior can be replicated in virtual reality; however, the device is grounded, limiting the user's freedom.

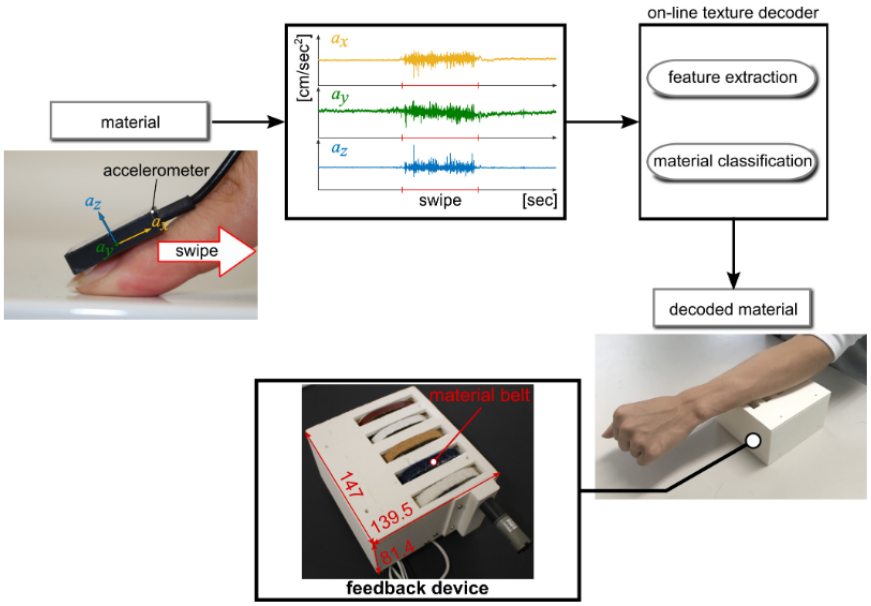

Figure 2.9: Sakurada et al.'s device and the different stages of material recognition [16].

[2]Merriam Webster's dictionary (accessed on 15/Nov/2022): https://www.merriam-webster.com/

Germani et al. [9] propose an approach based on electro-tactile stimuli to mimic textures. To correctly recreate a texture, first, the device evaluates the user's mechanoreceptors characteristics so that the intensity can be adjusted. When evaluating these characteristics, it is also noted how the distance the finger moved affects the perception of the texture. After enough information about the user is extracted, the device can provide the user with the correct intensity of electro-tactile feedback to correctly simulate the desired texture. Additionally, to further enhance the experience, auditory cues are also played, which proved to increase the ratio of correct classification of the material being mimicked. The final result can be observed in Fig. 2.10.

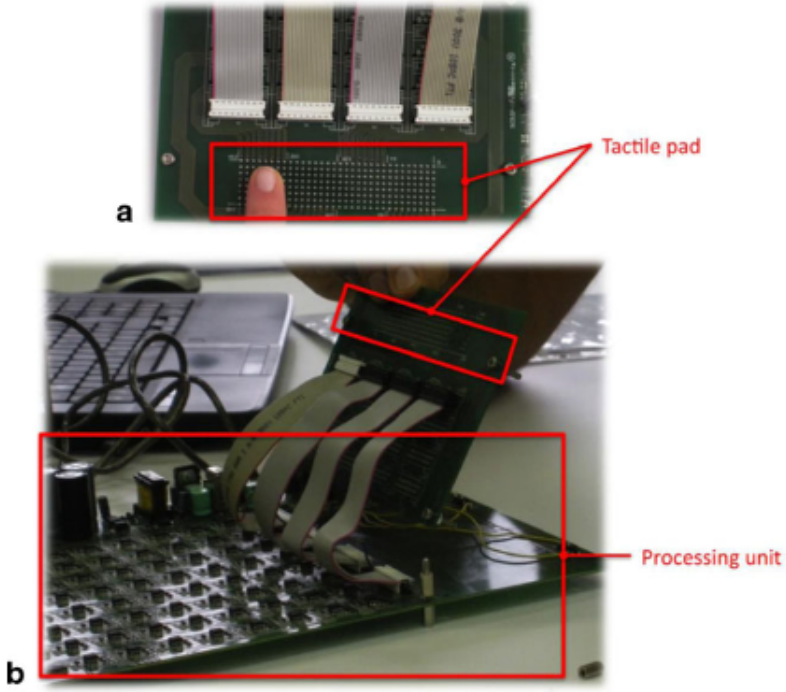

Figure 2.10: Germani et al.'s device, consisting of a pad to provide feedback (a) and a processing unit (b) [9].

DualVib [21] simultaneously provides two types of haptic feedback, represented in Fig. 2.11. The first is feedback created by vibrations that simulate the mass of an object moving around. The second is texture feedback through vibrations to mimic the texture and material of the object in the device's surface, taking into account the object's mass. When interacting with an object with free-moving mass inside, it is expected to hit the object's shell. With this in mind, *DualVib* creates vibrations on the device to mimic this free mass, forming different textures the user can feel.

2.3.2 Material Manipulation

Although most were grounded, there are some examples of ungrounded devices, such as the *HapticRevolver* [24]. Although this device is ungrounded, it is not without its limitations. The biggest one is that only one fingertip can feel the desired texture. *HapticRevolver* is a device with a handle

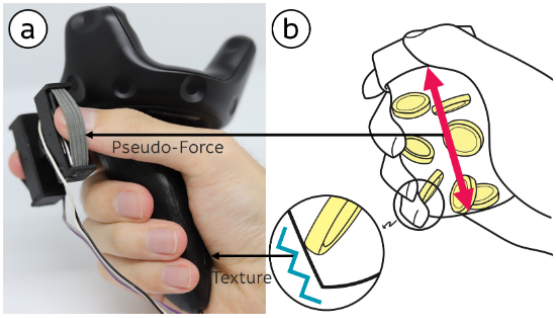

Figure 2.11: *DualVib*'s two types of haptic feedback [21].

where the user holds it and a spinning wheel with different textures where the index finger should be placed. This configuration can be seen in Fig. 2.12. This wheel then spins around to render contact with the correct surface texture. On top of that, it also provides the users with friction feedback in case the user is moving their hand through the virtual object.

Figure 2.12: Configuration of *HapticRevolver* [24].

Snake Charmer [1] uses a robotic arm paired with a head-mounted display camera to bring what is virtual to reality. Using the camera's reading and position, the arm can move an object, usually a geometrical structure containing a different material on each surface, to the place where the user interacts with the virtual object and rotate it to match the correct material, as seen in Fig. 2.13. By detecting the user's grasping force, it is possible to set the device to passive mode, allowing the user to dislocate the virtual object and the robotic arm. Finally, on top of recreating materials, it also simulates temperature through Peltier Pads.

Other explored solutions, such as the solution proposed by Günther S. et al.[11] and the solution proposed by Cai S. et al. [3], are present in section 2.3.3. Günther S. et al. proposes a device that strokes the users' arm with different textures while presenting visual feedback and evaluating the effect of different visualization for the same texture. Cai S. et al. proposes a deep-learning-based approach, that by taking an image of a surface, it's able to generate acceleration signals for the stylus-based input device.

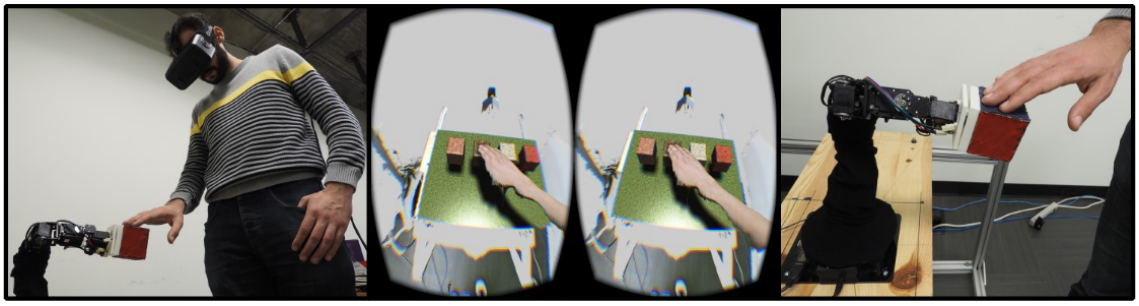

Figure 2.13: *Snake Charmer* with a cube attached [1].

2.3.3 Discussion

Every approach discussed in the previous sections tries to achieve the same goal: increasing the user's immersion in virtual reality. But even if the goal is the same, the way the devices achieve this is different. In this section, we will discuss approaches taken to highlight better some characteristics that can be used moving forward.

A summarized view of the devices and what they offer can be seen in Table 2.1. A device is classified as: *Graspable* if the user can grasp it; *Untethered* if the user can move it around freely and can put it down; *Shape-Changing* if the device changes its shape with the goal of making the users perceive this change; *Weight-Changing* if the device simulates weight changing or has a dynamic center of mass; *Material-Changing* if it can change its surface material or texture or mimic it correctly.

With a simpler view of the devices, we can see that very few devices try to mimic more than one characteristic at a time. Those that mimic more than one thing instead lack another characteristic, such as the lack of graspability and not being untethered like *ReCompFig*, or *Grabity* not being untethered. We aim for a completely untethered device as well as a graspable one, like *Shifty* and *PoCoPo*, since being untethered and graspable are critical aspects for the user's immersion.

By analyzing the table, it is possible to see that both *Shape Changing* and *Weight Changing* fields have made big advances in their respective area, having multiple *Graspable* and *Untethered* devices.

However, we can also see that material and texture-changing is the paradigm lacking the most. From the approaches studied, only two are graspable, and only one is untethered. We found this worth exploring and focused our efforts on the material and texture-changing paradigm, since the other paradigms are already more researched.

Almost none of the proposed solutions for material and texture-changing approaches try to provide feedback to the whole hand. Most of them focus on the fingertip, and although it is commonly used to feel textures, we decided to aim for a bigger contact surface. Compared to this, shape-changing devices do not have this problem. If we can adapt some mechanisms used in these solutions to our context, we should be able to create a device that provides material and texture feedback to the whole hand at the same time.

Table 2.1: Characteristics of haptic devices

Device	Graspable	Untethered	Shape Changing	Weight Changing	Material and/or Texture Changing
Shifty [27]	●	●		●	
Transcalibur [17]	●	●		●	
ElastOscilation [22]	●	●		●[3]	
Grabity [7]	●		●	●	
ReCompFig [25]			●	●[3]	
XRings [10]	●	●	●		
Eisuke Fujinawa et al. [8]	●	●	●		
PoCoPo [26]	●	●	●		
Ninja Track [13]	●	●	●		
Haptobend [14]	●	●	●[2]		
Elastilinks [23]	●	●	●		
Shape-a-Ghetti [2]	●	●	●		
DualVib [21]	●		●[1]		●[4]
Cai S [3]					●[4]
Germani [9]					●[4]
Sakurada [16]					●
Snake Charmer [1]					●
Steel Wool [11]					●[5]
Revolver [24]	●	●			●[5]

[1] Able to simulate simple shapes by deforming the surface the user holds, but the main goal is texture simulation
[2] Requires user intervention
[3] Through the use of inertia
[4] Only texture
[5] Only simulates texture and material on a specific body part

2.4 Summary

In this chapter, we explore some different paradigms used for haptic feedback in VR. We also analyze some different characterization of these devices, such as if the device is untethered and/or graspable. The devices studied were also separated into three different categories, these being: shape changing; weight changing and material and/or texture changing.

After analyzing the distribution of the solutions, it was noticed that material and/or texture changing solutions were the paradigm lacking the most. With this analysis, the focus of the work was defined, and we could start planning how to approach the problem.

Chapter 3

Creation of the Materializer

After analyzing the existing solutions for the various paradigms, we can start planning the solution to be developed. As stated previously, the solution must be able to mimic different materials and textures and also should be graspable and untethered.

3.1 Materializer: Designing The Material Changing Device

We first needed to settle on a design for the device to achieve this objective. The design should be simple enough so that it is easy to maintain and change the surface's material. After agreeing with the design, we need to decide how to control it and communicate with the application. We can start developing the required software to test.

All these stages of development are detailed in the following subsections by order. The first one will contain the solution's specifications. A concept will follow this for the solution, the resources needed to build this proposition, and the communication methods used.

3.1.1 Requirements Specification

Before considering how the solution would be implemented, we must first decide what the requirements. As previously stated, the final solution must be able to mimic different materials without having to change the device being used. On top of that, the device should also allow for an easy material exchange if the users so desire.

The following requirement is that the device serves as an input system for a virtual reality application. The user must be able to use the proposed solution as a way to interact with the virtual environment and the virtual objects in it. This requirement forces the device to be untethered, as a tethered device would limit the user's movement, hindering immersion. Alongside being untethered, the device should also be graspable so that the users can pick it up with one hand and move it as they see fit.

As such, the requirements the solution has are as following:

- **Be able to change material**: this is the main goal of the project. The device must be able to adapt to the environment, and to do this, the device must be able to change its material.
- **Be graspable, untethered, and ungrounded**: since the solution is to be used in VR as an input system, the device must be able to be moved freely. The user must be able to pick up the device, and move around with it, without being bound by anything.
- **Must be trackable inside the virtual application**: in order to be used as an input system, the VR application must be able to know the device's location in the real world.
- **Easily be able to swap materials**: in order to be used in multiple applications, the materials on the surface of the device must be able to be swapped easily. This makes swapping materials a trivial matter, instead of needing to create a new device entirely.

3.1.2 Concept

With all the requirements in mind, a consensus on the solution has been reached. The solution will have a design similar to *Trivision billboard* [1], where each side of every prism has a different material. A simple example of a *Trivision billboard* can be seen in Fig. 3.1. Differently to *Trivision billboards*, the prisms to be used in the final device are arranged circularly. This change was made so that when the user holds the device, it feels rounder than if every prism was aligned in a line. With this alteration, we can get a rounder surface for the user to grasp instead of a wider, planar surface.

We decided to use this mechanism, since it has been used in a real world scenario and proved to be effective at doing what we intend the device to do. The mechanism also proves generic enough, so that it can be adapted to a more round structure, while maintaining the functionalities a flat surface has.

The device supports an *HTC tracker* on top, so the application can know its position, thus using the device as an input system.

The prisms have to move synchronously so that the whole surface of the device has the same material. A structure similar to *Shape-a-Getti* [2] was created, with the main gear responsible for rotating every prism. Contrary to *Shape-a-Getti*, a brake to stop the device from changing its shape due to the user's grasping force will not be needed, as our device will have a filled surface. As such, the user will not be able to apply force in order to rotate the gears easily. This structure was selected for the final device, as it only uses one actuator and is able to achieve the main goal of rotating every prism synchronously.

Finally, the different components of the device, such as the base, the prism, and the top of the device will have to be independent of each other. This allows for a simple dismantling of the device, which provides the users with a simple way to swap the prisms that constitute the device.

[1] Trivision (accessed on 22/11/2022): https://en.wikipedia.org/wiki/Trivision

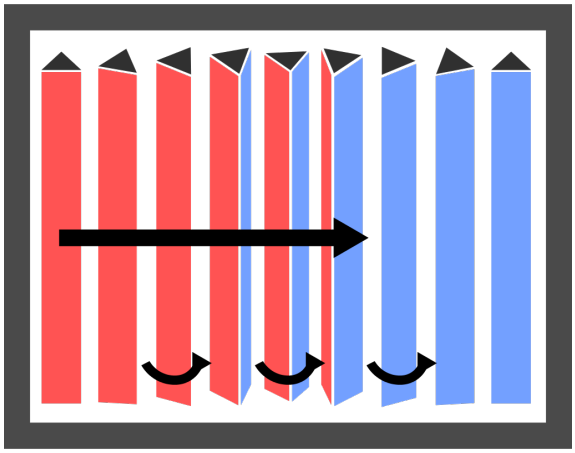

Figure 3.1: A simple example of a Trivision mechanism at work.

A simple prototype has been drafted using *Tinkercad*[2] for easier visualization, and can be seen in Fig. 3.2. This design uses prisms with plain surfaces where the different materials would be glued to.

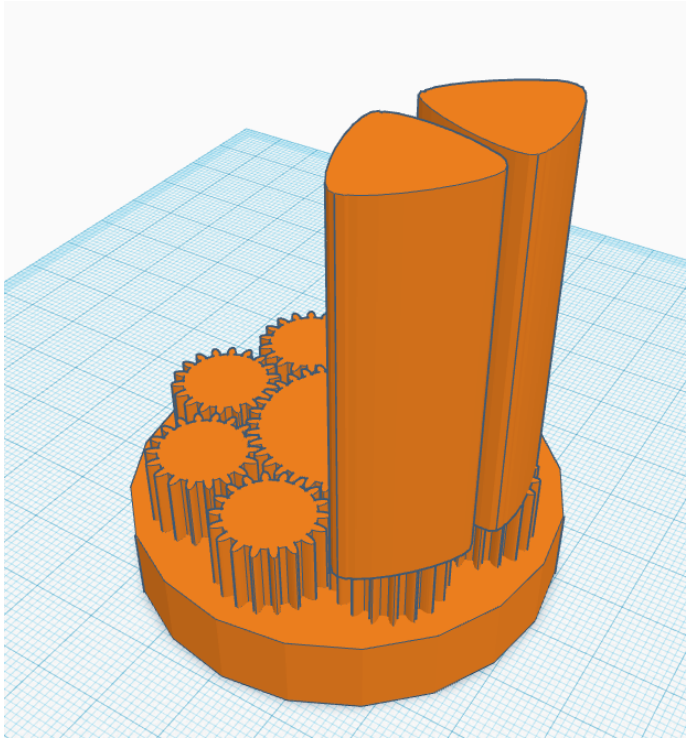

Figure 3.2: Initial design of the device using the *Trivision* mechanism.

[2]Tinkercad (accessed on 1/1/2023): https://www.tinkercad.com

The main gear would rotate the desired amount, which in turn would cause the smaller gears to rotate all at the same time.

3.1.3 Needed Resources

In order to create the wanted models, an SLA 3D Printer was used, more specifically, the *SLA model ELEGOO Mars 3 resin 3D Printer* [3]. To modify the device, a single servo motor is used which is controlled by a Microprocessor. Initially, this Microprocessor used a cable connected to a power supply as an energy source, but in order to create a truly untethered device the power source was changed to portable batteries and communication to *Bluetooth*. All the gears needed, in order to make the prisms rotate, were also 3D Printed. Lastly, an *HTC Vive* tracker is also needed in order to obtain the device's position inside the VR application.

3.1.4 Summary

In this chapter, we presented the main idea behind the device created. The device follows the *Trivision* design principle, but adapts it to a more cylindrical structure. The device contains a main gear, which when rotated, makes every other smaller gear rotate synchronously. These gears are connected to a prism containing different materials in each face. Lastly, the needed resources are also detailed in this chapter.

[3] 3D printer used (accessed on 1/1/2023): https://www.elegoo.com/products/elegoo-mars-3-lcd-3d-printer

Figure 3.3: Side by side view of Materializer and Shape-a-Getti.

3.2 Materializer: Prototyping the Material Changing Device

With the requirements defined, we started creating the device's structure.

The overall design of the structure was based on *Shape-a-Getti* [2]. The experience gathered by the team during the creating of *Shape-a-Getti* was essential to the development of *Materializer*, and allowed for a faster development of the device. A side by side view of the two devices, *Materializer* and *Shape-a-Getti*, can be seen in Fig. 3.3.

The device has three main components, which are the base, the prisms, and the roof. The prisms are used to support the materials in the device, rotating synchronously to display the same material in the device's surface. The device is supported by a metal rod that connects the base to the roof of the device. The prisms are held in place using the holes in the roof where the prisms connect with it. The components used to rotate the prisms are held inside the base. The base contains a set of gears connected to a main gear, making them rotate synchronously. The main gear is rotated using a motor also present in the base.

3.2.1 Base Lid

An upgrade that needed to be done was to the base lid. The lid used for *Shape-a-Getti* was very thin, which made it fragile. Said lid ended up breaking in the testing phase of *Shape-a-Getti*, so it needed to be printed once again. Taking advantage of this, the team decided to make an upgrade to the lid and increase its width, making it more resistant. The newly printed lid can be seen in Fig. 3.4. The original design and dimensions of the lid was kept so the remaining pieces could still be used.

Figure 3.4: Upgraded lid component with thicker connector component

3.2.2 Prisms

When preparing the device to be printed, some changes were made. The main one was the way the material got added to the surface of the prism. Initially, the material would be glued to the surface or mounted on it. This method of applying the material could cause problems, as it would make iterating the design harder. Every time a new prism design was printed, a new piece of the material would need to be acquired.

In order to uncouple the material from the prism, an updated design was created. This design, as can be seen in Fig. 3.5, contains a subsection in the prism into which the materials can be slid. This new way of applying the material makes changing the prism design easier, since the same material can be used repeatedly. A closer look at the gap where the material can slide into can be seen in Fig. 3.6.

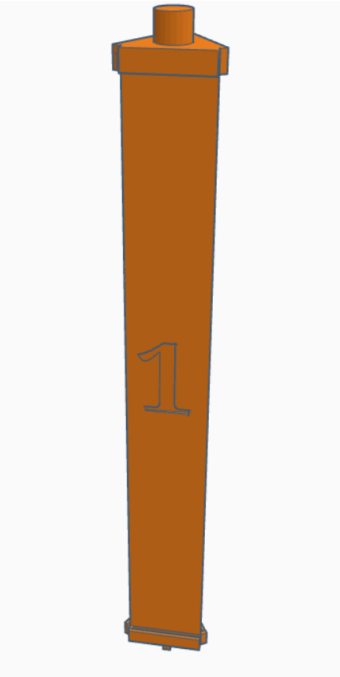

Figure 3.5: Updated design of the prism used in the device.

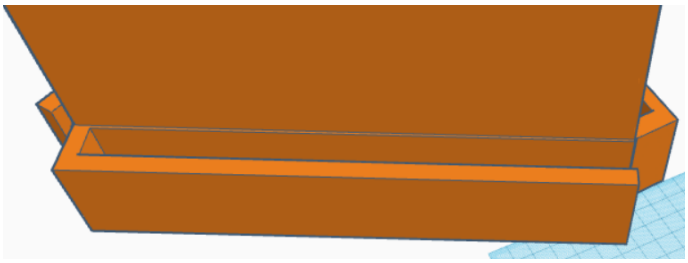

Figure 3.6: Closer look at the gap where the material slides into.

On top of making updates to the design more efficient, it also makes changing materials easier. Instead of having to obtain multiple new prisms in order to mimic another material, this new material can be swapped with one being used.

The prisms used in the device were all 3D printed and designed precisely for the device being developed. A total of 5 prisms were used in the final design, and an example can be seen in Fig. 3.7. The prisms were created using *Tinkercad* [4]. These prisms rotate in place, being able to display to the outside a different material with each rotation.

The prisms all have a *LEGO Technic* axel [5] shape at the bottom of it, so that they can be connected to the outer gear of the base. This axel can be observed in Fig. 3.8. On top of that, to make this piece more robust, a cylinder was made to wrap around it near the prism, since it seemed like a weak point where multiple prisms broke when force was applied. The top of the device has a cylinder that connects to the roof, so that the prism can be held in the same place when rotating

[4] Tinkercad (accessed on 1/1/2023): https://www.tinkercad.com

[5] LEGO Technic axel (accessed on 30/5/2023): https://www.bricklink.com/catalogList.asp?catType=P&catString=134

Figure 3.7: Image of a prism used to hold the materials.

Figure 3.8: *LEGO Technic* axel at the bottom of the prism.

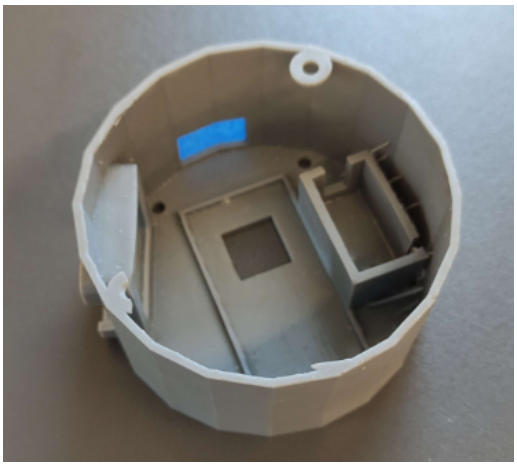

Figure 3.9: Empty base used to hold the hardware

Figure 3.10: Base filled with hardware

3.2.3 Reutilized Pieces

The other pieces that make up the device, more precisely the gears, the base that holds the microcontroller and the device's ceiling are the same used for *Shape-a-Getti*. All new pieces were designed with the specification of these pieces, so that by doing so, development could be faster by reutilizing these pieces.

The device's base was designed to be separated into three different parts. The bottom part is used to hold a battery and a motor that works as a break. For the purpose of the device developed in this dissertation, the break was not needed, as the force done by the users does not conflict with the motor. Finally, the base also has a place on the side that fits the M5Stack, exposing the USB ports to the outside. This base can be seen empty, as well as holding the components in Fig. 3.9 and 3.10, respectively.

The middle part of the base holds the gears used for the device's rotation. There is a middle gear, the main gear, which has a diameter of 45.32 millimeters (teeth included) and a total of 60 teeth. The smaller outer gears have a diameter of 10.23 millimeters and 12 teeth. This means that for a rotation of 1º in the main gear, each small gear rotates 5º. The main gear can be seen in Fig. 3.11, and a small gear can be seen in Fig. 3.12. The third part serves as a cover for the base, which was printed anew for this device. The base was connected and held together by a group of four M3 screws. In Fig. 3.13 it's possible to see the base connected by said screws without the lid, with the gears in place.

The final piece that is used in the device is its roof, which can be seen in Fig. 3.14. This piece contains holes where the prisms' upper cylinder connect. There is also a hole in the of the roof so that the rod that holds the device together can go through. This rod is a M6 metal rod that is used to connect the device's base and roof, in order to give stability to the device. The rod is also used to screw a *HTC Vive* tracker to the top of the device, in order to know the device's position inside the virtual environment.

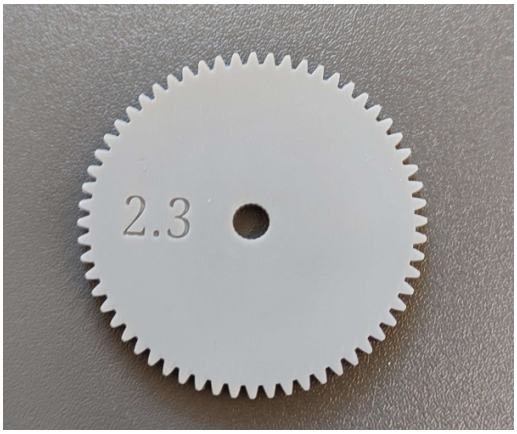

Figure 3.11: Main gear.

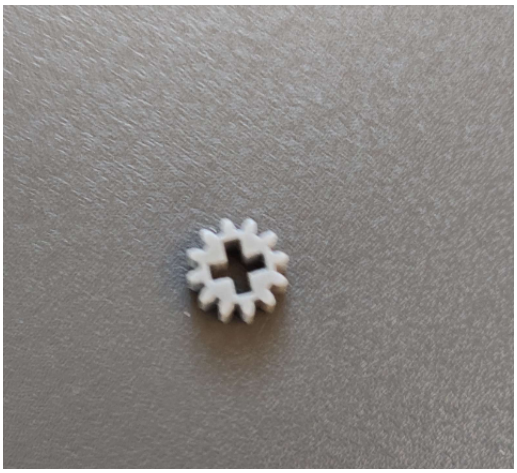

Figure 3.12: Small gear.

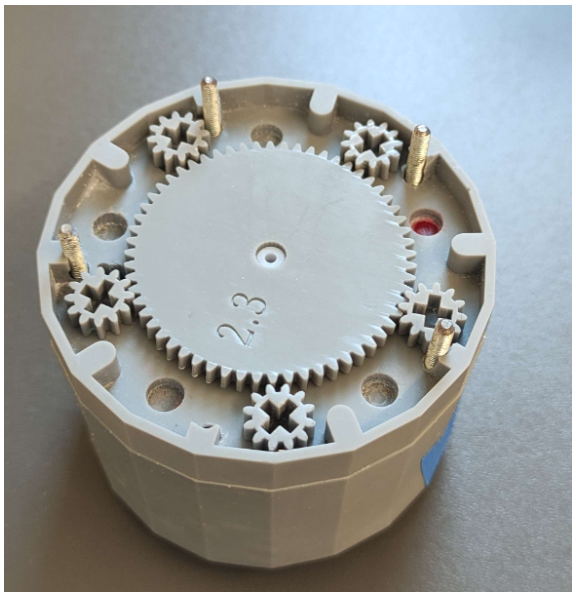

Figure 3.13: Base without the lid.

3.2.4 Final Device

The final device can be seen in Fig. 3.15 next to a 500 ml bottle of water for comparison. The device is 213 mm high, from base to roof, and the base has a diameter of 50 mm. An exploded view of the device wade in *Tinckercad* can be seen in Fig. 3.16.

Limitations

Sometimes the outer gears were so far from the main gear that these gears would not rotate. This distance would cause some configurations where the surface of the device was not homogeneous. To mitigate this problem, a rubber band is used to join all small gears of the prisms with the main gear. This rubber band can be seen in Fig. 3.15 as the red colored line at the bottom of the prisms.

Figure 3.14: Roof used for the device.

If the users held the device near the roof, sometimes this would cause the roof piece to rotate. This would misalign the small gears from the main gear, making the device's surface not alter. The nut that holds the roof in place was tightened as much as possible to try to erase this, but we could only reduce the frequency of this.

Due to time constraints, these two problems were not explored further. Due to the long print times, printing new pieces to erase these problems would hinder the testing phase, making the results obtained and analyzed less desirable.

3.2.5 Summary

This subsection details every piece used to build the device. It contains which pieces were 3D printed anew, and the ones that were reused from previous prints. An explanation for each piece's functionality is also provided. An overall view of the device is done, detailing how the device works, as well as the device's dimensions, are contained in this subsection. Finally, this subsection is closed by pointing out some shortcomings with the design, and what was done to fix some of them.

Figure 3.15: Device next to water bottle.

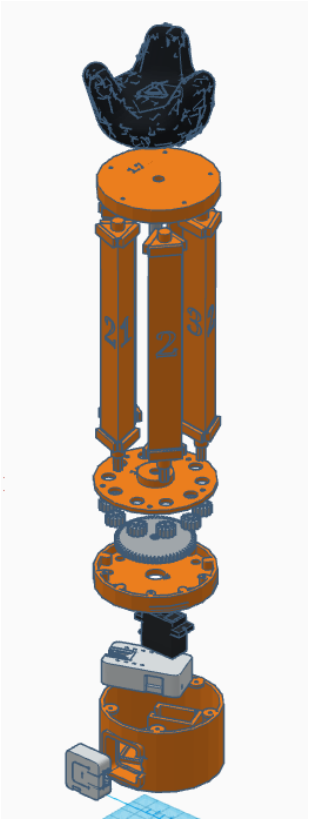

Figure 3.16: Exploded view of the device.

3.3 Materializer: Controlling the Device

One of the most essential parts of a material-changing device is how the application can manipulate said device. The application that controls the device must be able to correctly identify which materials are on each surface of the prisms, as well as the rotations needed to obtain the correct device configuration. This means allowing communication between the VR application and the microcontroller.

At first the communication was being done exclusively by serial port, but during later stages of development the communication changed to *Bluetooth*. The microcontroller used also supports Wi-Fi connection, however a *Bluetooth* is enough, since the device is to be used close to the computer where the VR application is running.

3.3.1 Architecture

The architecture for the solution can be found in Fig. 3.17. The Unity[6] software communicates with the microcontroller, which is an M5Stack ATOM Lite [7], wirelessly or through a physical serial port, and receive information from the tracker wirelessly. The wireless communication with the M5Stack is done through *Bluetooth*. The M5Stack is responsible for taking the information received from the application, and to control the motor that changes the configuration of the device. The application was written using Unity, as it provides a simple engine to create and develop applications for virtual reality.

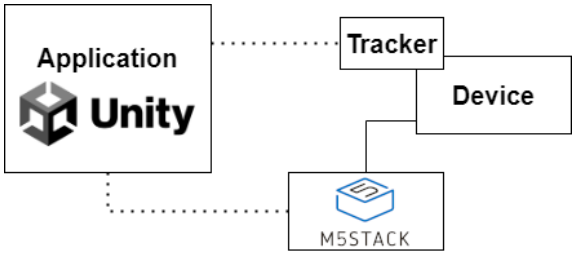

Figure 3.17: Architecture of proposed solution

3.3.2 Microcontroller

The microcontroller used is an M5Stack ATOM Lite. This microcontroller was initially programmed using the Arduino IDE [8]. This provided an easy way to install and upload any necessary library, as well as an easy way to burn the code into the microcontroller.

When implementing the *Bluetooth* communication, the development switched from Arduino IDE to PlatformIO [9] inside Visual Studio Code. This new environment provided access to new

[6] Unity (Accessed on 9/1/2023): https://unity.com/
[7] M5Stack (Accessed on 10/3/2023): https://m5stack.com/
[8] Arduino IDE (Accessed on 10/3/2023): https://www.arduino.cc/en/software
[9] PlatformIO (Accessed on 29/5/2023): https://platformio.org/

libraries which were crucial to the *Bluetooth* implementation, more specifically, *BluetoothSerial*[10]. The microcontroller has an RGB LED that was used for debugging purposes. While not connected to the VR application, the LED is green. When the microcontroller establishes a connection to the application, the green LED is changed to a blue color.

When waiting for a connection, the device appears as *"Material Changer"* to any nearby *Bluetooth* device. When the connection is established, the device is ready to start receiving messages. If the message received is unknown, nothing happens and the microcontroller stays idle waiting for the next message. The code uploaded to the microcontroller is responsible for receiving the messages sent by the unity application and parsing them in order to manipulate the device. The messages come in the following format: "R=XX", where R symbolizes the rotate command, and XX is the amount the motor must rotate. For example, the message "R=60" makes the motor rotate to where it is at 60 degrees. A diagram of the whole process, from receiving the message to sending a message to the motor, can be seen in Fig. 3.18.

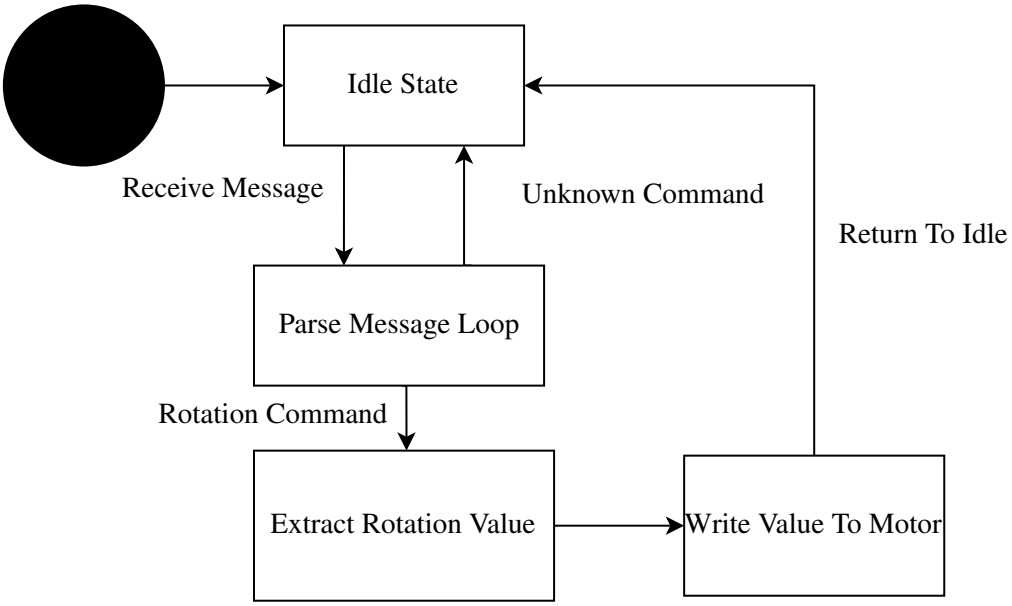

Figure 3.18: Microcontroller's process diagram.

The microcontroller is responsible for receiving the messages sent by the Unity application, and splitting it so that it's possible to control the motor and rotate it to the desired angle. A simple serial event handler was implemented, that upon receiving a message, the message is parsed, and the motor is rotated to the desired angle. Other solutions were explored, more specifically, "SerialCommand"[11]. This library allows for easier parse of multiple distinct messages, but since the microcontroller only has to interpret one type of message, which is the "R" message, it was deemed not needed and a simpler communication method was used.

[10] *BluetoothSerial* (Accessed on 29/5/2023): https://registry.platformio.org/libraries/mbed-components/BluetoothSerial

[11] SerialCommand (Accessed on 17/03/2023): https://github.com/argandas/SerialCommand

The motor being used is a M5Stack 180º servo motor. This motor is only able to rotate 180 degrees, however, since the ratio between the main gear and the gears used to rotate the prisms is bigger than one, 180 degrees is enough for every configuration.

The code used to handle the messages received and to rotate the device can be seen in the Appendix C.

3.3.3 Unity Application

The virtual environment was developed using Unity. This development platform provided easy access to frameworks to facilitate virtual reality application development, such as the *Steam VR* framework [12]. This framework allows for VR simulation using keyboard commands, which helped debug the application without using the VR HMD. Using the keyboard to simulate the controls for the virtual reality application allowed for a faster and more efficient development of the application, as setting up the VR system was not needed every time the application was updated. On top of that, it allowed for the development of the application in more places, rather than having to do it strictly at the lab.

In order to make the application as generic as possible, every piece of information about the materials used is passed through a *JSON* file and interpreted at the start of the application's execution. An example of how the information is passed to the application can be found in the snippet 3.1. The actual values of the *deviceRotation* field may vary, depending on which material is placed facing out when assembling the device.

Listing 3.1: Example of JSON to provide information about the materials.

```
{
  "objects": [
    {
      "name": ["metal", "stone", "glass"],
      "deviceRotation": 0
    },
    {
      "name": ["wood","cork"],
      "deviceRotation": 24
    },
    {
      "name": ["foam","sponge"],
      "deviceRotation": 48
    }
  ],
  (...)
}
```

[12] Steam VR framework (Accessed on 22/05/2023): https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

The application receives an object with a key *objects*. This key corresponds to a list of objects, each with keys *name* and *deviceRotation*. The key *name* has a list value that contains a string of every material's name. This list groups the virtual materials represented by the same real-world material. The key *deviceRotation* represents the rotation the motor needs to be at so that all prisms have the correct material facing out. Initially, every material had an entry in the *JSON* file instead of being grouped in a list. While easier to handle initially, this caused some problems later on in the second task, as knowing which materials belonged to the same group was crucial. This way, instead of having to check the hey *deviceRotation* and grouping every material, the materials come grouped right away from the *JSON* file.

The *JSON* file also contains the order in which the materials should be guessed by the users in both tasks. This is contained in two list with a shared parent, one for each task, where the order of the list, is the order in which the materials need to be assessed by the users in each task. The order in which the materials need to be assessed follow a *latin squares* distribution. This distribution is explored further in subsection 4.4.

In order to communicate with the microcontroller, a simple communication via serial port is made. In order to establish a connection and handle the exchange of messages, the System class, *System.IO.Ports* [13], was used. This library contains classes for controlling the serial ports. Since the *Bluetooth* connection creates a virtual serial port between the computer where the application is running and the microcontroller, this library can be used for both the physical and the *Bluetooth* connection. This communication method is used when the application detects the user is interacting with a virtual object. When detecting this interaction, the *Unity* application sends a message with the format explained in the previous subsection, making the device alter its surface material.

To obtain the device's location, the class *SteamVR_TrackedObject* was used. This class is contained in the *Steam VR* framework. The inspector view of the class can be seen in Fig. 3.19. This class has a field called *Index*, which must be filled with the number associated to the tracker. This index number was obtained though trial and error, as we did not find a way to determine it before running the application. This index number would also sometimes change when the tracker was connected to the *HTC Vive*'s link box.

Figure 3.19: Inspector view of the *SteamVR_TrackedObject* class.

[13] System.IO.Ports (Accessed on 4/4/2023): https://learn.microsoft.com/en-us/dotnet/api/system.io.ports.serialport

3.3.4 Summary

In this chapter, we describe how the device is controlled. The device is controlled using a microcontroller, that is able to connect both via a physical serial port, or through *Bluetooth*. The device receives messages from a *Unity* application, rotating the motor to the desired rotation angle. The *Unity* application obtains the information about the angles through a *JSON* file, whose structure is also detailed in this chapter.

Chapter 4

Evaluation

The goal of this thesis is to create a more immersive way for the user to interact with the virtual objects around him, simulating materials that the user encounters in said environments. Because of the nature of these types of devices, user validation is a must.

Since the device was designed to change its surface material, the tests' main objective is having the users correctly identify what materials are currently being represented by the device.

The tests were conducted to attempt to obtain the answer to the following questions:

- Are the users able to correctly identify the materials being presented on the device's surface?
- Can the same material on the device's surface represent similar materials in the virtual environment with the help of visual cues?

4.1 Procedure

The experiments occurred in GIG's (Graphics, Interaction and Gaming) lab at FEUP. The lab is located in the "I" building in room I220. The lab provided the VR equipment and space needed to perform the user tests.

Before interacting with the device and the equipment, the users needed to fill a form containing some basic information about themselves. The form also contained some questions about how accustomed the users regard the use of VR technologies. Finally, the users were asked to consent to have their experience's audio recorded so that it could later be analyzed and some photos taken. All the questions and statements used in the form can be found in the Annex B.

After filling out the form, we guided the users to the allocated space in the lab where the VR headset could be used. Before putting on the headset, the users did not see or analyze the device. Not showing the device was done, so the visible materials did not influence the users on the device's structure.

We expected the experience not to take over fifteen minutes. In practice, the experience took an average of 7 minutes and 26 seconds, the slowest taking 14 minutes and 27 seconds and the fastest taking 4 minutes and 50 seconds to complete. This variance originated in the time each

user took to analyze the device and to choose the material, as some users took longer to analyze the multiple materials than others.

After the users completed the experiment, they were asked to answer a post-experiment form. This form contained three statements that the users rated, which were:

- "I felt that the different materials improved immersion."
- "I could easily distinguish which material corresponded to the virtual object."
- "I felt approximately this many different materials."

The first statement's goal is to assess if the device truly helped with the users' immersion, or if the device was not necessary for the users to feel present in the environment. The second statement was used to verify if any virtual object felt out of place to the users, with no physical representation. Finally, the third question was essential to know if the physical materials could replace different materials in the virtual environment. This was only possible since the users did not have a chance to see the device before the experiment began and, as such, did not know the device only had three different materials. The first two statements were answered using a five point *Likert scale* [1], while the third statement was answered with a number.

4.2 Selected Materials

Since the device was created to hold three different materials, a search for distinguishable materials was done. According to [28], there are multiple materials that can be recognized by haptic and visual feedback. These are the materials that were used to test the device, as the device replicated the material that is displayed in the virtual environment. The three selected materials were *metal*, a *wood* and finally *foam*. All these materials can be identified both visually and with haptic feedback.

In order to try to mimic different materials with materials already present on the device, we need to use materials that are similar to the ones being used by the device and that can be easily identified by visual feedback. This visual feedback may lead the users to believe that they are interacting with a material different to the one they really are interacting with. To find these materials, some materials were classified according to the properties that can be found in Elvin Karana's et al. [12]. The characterization of the properties can be found in table 4.1. Some *Tactual* properties were not included in the analysis, such as weight and ductility, since these two properties are not important for this work, and have some research done on their own specific field. On top of that, only the *Tactual* properties of the material were analyzed, since the *Visual* properties can be created inside the virtual environment, and the *Olfactory* properties are not important for the evaluation.

Although multiple materials were analyzed, only a few needed to be selected. The table helped select some materials that can be used to verify if, with the help of visual feedback, the same

[1] Likert scale (accessed on 15/06/2023): https://en.wikipedia.org/wiki/Likert_scale

Table 4.1: Characterization of different materials.

Characteristics		**Pressure**		**Force**				**Friction**		**Temperature**	
		Softness		**Strength**		**Elasticity**		**Roughness**		**Warmth**	
		Hard	Soft	High	Low	High	Low	Rough	Smooth	Hot	Cold
Metallic	**Metal**	●		●			●		●		●
	Stone	●		●			●		●		●
	Glass	●		●			●		●		●
Wooden	**Raw Wood**	●		●			●		●	●	
	Compressed Cork	●		●			●	●		●	
Foam	**Foam**		●		●	●		●		●	
	Sponge		●		●	●		●		●	
Other	**Processed Wood**	●		●			●	●			●
	Natural Cork		●			●		●		●	

material can mimic multiple different materials, while cutting some that were not necessary. The selected groups of materials are:

- Metallic Group: *Smooth stone* (which will be referenced as *stone*), *glass* and *metal*, which have similar tactile characteristics. Although their color is different and glass is transparent, these are two characteristics that can be adjusted in the virtual environment.
- Wooden Group: *Compressed cork* and *raw wood* have a similar feel to the touch (which will be referred to as *cork* and *wood* throughout the rest of the document), and whose visual characteristics can also be adapted in the virtual environment.
- Foam Group: *Sponge*, which has a similar feel to *foam*, being the biggest difference in their density and holes size.

The physical materials can be seen in Fig. 4.1, as strips with 137 mm of height, 18 mm of width, and 0.7mm of length. The device with *metallic* surface can be seen in Fig. 4.2, with *wooden* surface in Fig. 4.3, and with *foam* surface in Fig.4.4.

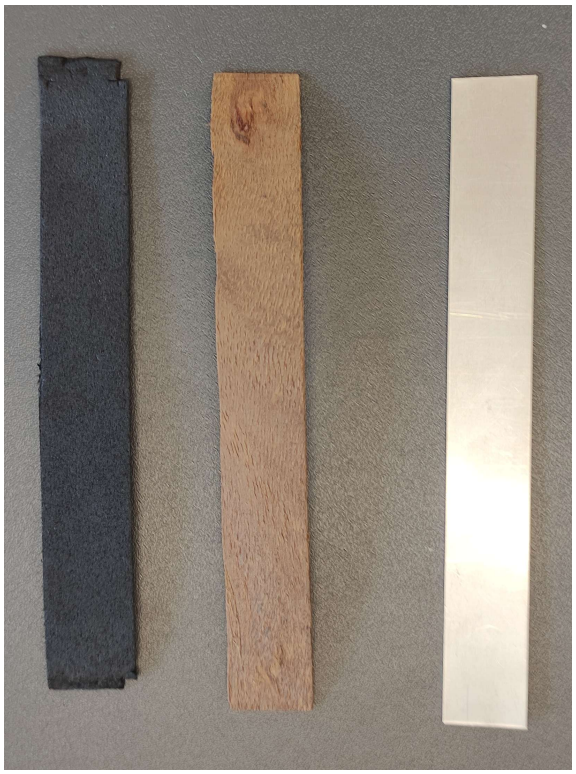

Figure 4.1: Material strips. From left to right, the materials are: *foam*, *wood*, and *metal*

Figure 4.2: Device with metallic surface.

Figure 4.3: Device with wooden surface.

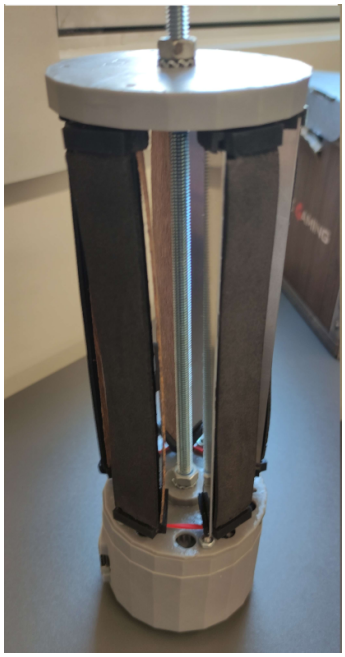

Figure 4.4: Device with foam surface.

4.3 Participants

A total of 21 (14 male, 6 female, and 1 preferred not to disclose) participants were involved in the experiment. The majority of the people that volunteered to test the solution were aged between 21 and 30 years old (12 people). The other participants belonged to many age groups, these being 16 to 20 (2 people), 41 to 50 (3 people), and more than 50 years old (4 people). Of the participants that tested the solution, 12 claimed that they had used virtual reality technologies before but not regularly, 1 said to use it weekly, and 8 claimed to have never used it.

4.4 Tasks

The users performed two distinct tasks, each repeated seven times. The first task involves displaying the user a group of objects consisting of a pre-selected object and two other random objects. The user needs to feel the device, whose surface adapts to the pre-selected object, and correctly assess the material being portrayed on the device. The randomly selected objects are filtered so that no two objects represented by the same real-world material are placed in the same group. Not representing objects with similar materials is done, so the user does not get confused by similar materials and selects the one not represented. Fig. 4.5 represents the physical, and virtual environment for the first task, if the represented material is *metal*.

Figure 4.5: Example scenario of the first task. Virtual environment on the left with three objects, physical environment on the right with a user interacting with the device.

The second task attempts to validate if the objects classified as having similar materials are perceived as such. This allows this type of classifications to be used as an indicator of which virtual objects can be represented by the same physical objects. The validation is done by displaying all seven objects at the same time together in a randomized order and asking the users to identify as many objects as they think the material being displayed by the device represents, thus creating groups of what the users think have similar materials. With the users made groups created, we can compare these to the groups created initially by the team. This allows us to see if the perception

Trial	Task 1	Task 2
1	G P W S F M C	P G F M W C S
2	C S P M W F G	G M W F C S P
3	M C F G S W P	W C G P S F M
4	S M G F C P W	F S P C G M W
5	P F M W G C S	C P S W M G F
6	F W S C P G M	M F C S P W G
7	W G C P M S F	S W M G F P C

Table 4.2: Order of simulated materials.
M - metal, P - polished stone, G - glass, W - wood, C - cork, F - foam, S - sponge

of the different virtual materials using the same physical material is in agreement with the groups classified before the tests. To facilitate the user's experience, each virtual object is labeled with a number from 1 to 7. An example scene of the second task can be seen in Fig. 4.6.

Figure 4.6: Example scenario of the second task.

In both these tasks, the users will be asked how confident they are about the choice made in a scale of 1 to 5. These confidence levels are expected to drop in the second task since the user might be confused about multiple objects, decreasing the confidence in their answer.

The task order can be found in the table 4.2. The order followed to represent each object was obtained using *latin squares* to minimize the bias users may have depending on the order in which the objects follow.

4.5 Setup

Although *Bluetooth* communication was implemented, the communication was done through a serial port, as the computer used for the tests did not have *Bluetooth* connections support and that factor was not relevant for these tests.

The physical setup can be seen in Fig. 4.7. Inside the cardboard box is the device the users have to interact with. Before the users place the HMD on their head, the box is covered with a towel, so that the user cannot see what it has inside. On the side of the box, there is a hole to allow for the cable to be connected with the microcontroller. The *HTC Tracker* is screwed on top of the box in order to obtain its position in the virtual environment.

Figure 4.7: Physical environment used for the tests.

4.6 Prototype

Since the device focuses on the object's material and texture, every object is represented by a cylinder to best match with the device's physical form. The materials are displayed on top of a table inside the virtual environment developed using Unity. The environment resembles a simple computer room, to be similar to the environment the users were in before putting on the headset.

Before and during the testing phase, the device was hidden inside a box to not influence the users choices. The box has a tracker, so its position can be correctly represented in the virtual environment to help guide the users. The virtual environment can be seen in Fig. 4.5 and Fig. 4.6.

The results were being recorded by the team, by pressing in the keyboard the number corresponding to the material chosen by the user, who would say the number out loud. After each guess done by the user, there was a countdown for the next trial, so that the device had time to adapt itself. On top of that, a sound is played to drown out the noise the motor produces when changing shape.

Figure 4.8: Virtual environment used for testing.

When the user tests were finished, by completing the tasks or having the team's facilitator pressing the "G" key, the results of the test were stored in a *JSON* file. This *JSON* file is the converted into a *CSV* file, ready to be analyzed.

4.7 Results

The results obtained from the two tasks were grouped together, and analyzed using *Pandas*[2].

4.7.1 Task 1

Fig. 4.9 represents a confusion matrix for the first task. The *Y-axis* corresponds to the material that was being represented by the device, while the *X-axis* corresponds to the material the users predicted. Each cell contains the percentage of correct predictions. As we can see by the matrix's main diagonal, most of the time the material being represented was correctly assessed, with the material *metal* having a 100% correct assessment.

On top of this, under each value is the average confidence level of the users reported while selecting the materials. The average confidence level for each correct answer (the main diagonal) is at least 4, showing that the users were confident in their answers. Although the confidence of the correct answers is high, there are some incorrect answers with high confidence. This may be because some users have a different perception of the same material (for example, some users claimed *"Certain materials, such as wood, may mean different materials depending on the type of wood."*), which may be the cause for the higher confidence in some incorrect answers.

[2]Pandas (accessed on 22/06/2023): https://pandas.pydata.org/

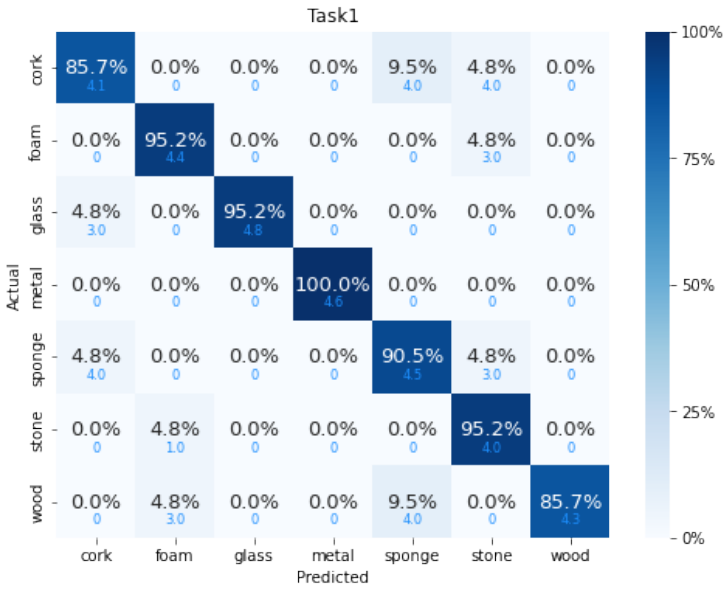

Figure 4.9: Confusion matrix for the first task.

4.7.2 Task 2

Fig. 4.10 shows a bar plot with the most common groups done during the second task. This graph only contains groups that were created at least two times and the groups are represented in percentage, since the metallic group had more guesses than the non-metallic group, 63 guesses for the metallic (21 for the materials *metal, glass*, and *stone*) and 42 for the non-metallic (21 for the group of materials *wood* and *cork* , and *foam* and *sponge* for the other group).

As can be seen in the plot Fig. 4.10, the three most popular groups are [*glass, metal*], [*cork, wood*], and [*foam, sponge*]. The two last groups, [*cork, wood*] and [*foam, sponge*], are representative of the groups made by us before the testing began, which is in agreement with the classification of these materials. The remaining group, [*glass, metal*], misses the material *stone* in order to be complete, which only about 6.4% of the guesses grouped these three materials. This correlation between *metal* and *glass* was also voiced by many users, with some claims such as *"Difference between metal and glass was not easy. And the same situation for sponge and foam."*, and *"Hard to distinguish between smooth surfaces like stone, glass and metal"*. Some users claimed they did not feel the *stone* material, as they were expecting a cold material with some texture instead of a material with no texture, as we classified beforehand.

The table 4.3 shows the percentage of people that got at least one correct answer. This means that an answer of, for example, *metal, glass* for a represented material of *stone*, is considered a correct answer, as these material belong to the same group.

By analyzing this table, we can see that the metallic materials were correctly assessed 100% of the time, although the *stone* material was mostly correct via its group materials. The other two groups, while not 100% accurate, have above 90% correct answers, the lowest being sponge and cork which are 90.5%. The average confidence level for these answers can also be seen in Table

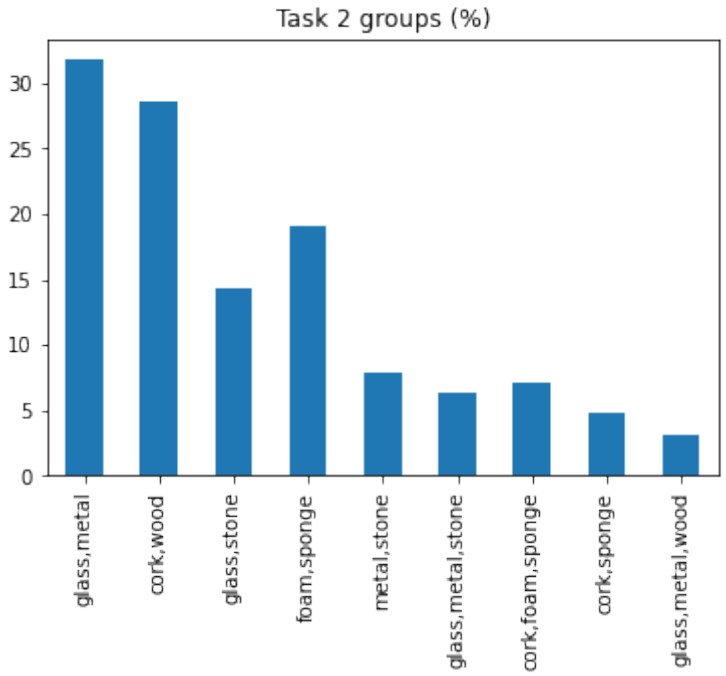

Figure 4.10: Most popular combinations for the second task.

4.3. In this table we can see that the average confidence for the answers with at least one correct answer is fairly high, most having a confidence level higher than 4, the only not following this being *glass*.

On the other hand, the table 4.4 shows the percentage of answers provided by the users that contained at least one incorrect answer. For example, if the user's answer is *metal, glass, sponge* and the represented material is *metal*, this answer contains at least one incorrect answer.

From analyzing this table, we can see that wood had the highest rate of at least one incorrect answer. From analyzing the results, most also answered foam or sponge as a material that was being represented (67%). This error rate in the *wood* material, may be due to the different perception the users have of the different types of wood. The same thing happened to the material *cork* and *foam*, where these three materials were sometimes mistaken for one another. Although there were

Table 4.3: Table containing the percentage of answers with at least one correct answer and the respective confidence.

Material	**At Least One Correct Answer Given (%)**	**Average Confidence**
Metal	100.0	2.0
Stone	100.0	2.0
Glass	100.0	4.0
Wood	95.2	3.7
Cork	90.5	4.0
Sponge	90.5	3.25
Foam	95.2	3.7

Table 4.4: Table containing the percentage of answers with at least one incorrect answer and the respective confidence.

Material	At Least One Incorrect Answer Given (%)	Average Confidence
Metal	4.8	2.0
Stone	4.8	2.0
Glass	4.8	4.0
Wood	28.6	3.7
Cork	19.0	4.0
Sponge	19.0	3.25
Foam	14.3	3.7

some incorrect answers between the correct ones, we still find the results obtained in this study to be satisfactory. On top of that, the answers with at least one incorrect answer had lower average confidence level, showing that the users were unsure in some of the groups they were forming. The average confidence level of the answers can also be seen in Table 4.4. The confidence for the material *glass* is higher when there is at least one wrong answer than when it had at least one correct answer. However, we can not conclude much from the *metallic* group, since only one person got at least one answer wrong for each element.

Furthermore, many guesses were made containing only one material. As can be seen in Fig. 4.11, over 50% of the *foam* or *sponge* simulations were answered with only *foam*, while only around 6.4% of the metallic group guesses were answered with only *stone*. The percentage of instances where only the materials of *metal* and *glass* were selected is relatively low, as these two materials were the most confused. This confusion leads the users to group them, as was observed in Fig. 4.10.

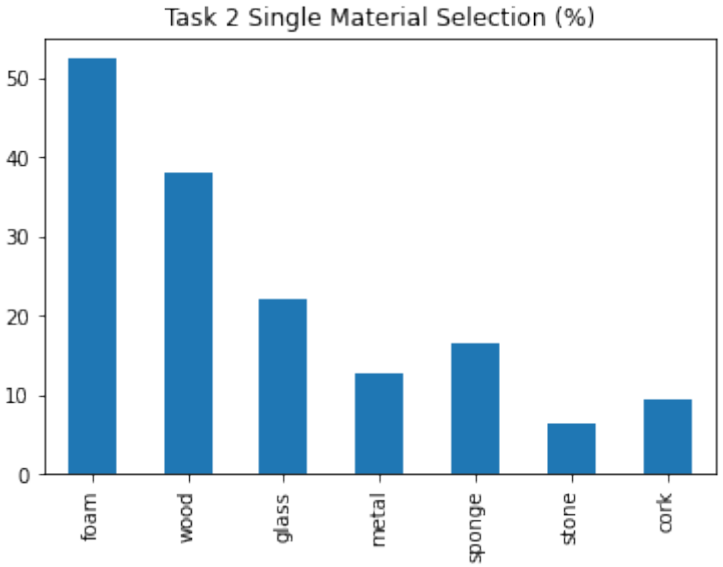

Figure 4.11: Percentages of guesses with only one material.

Contrary to what we initially believed, this second task did not alter the users' confidence

too much. When answering with a group of materials, the average confidence was 4.0; when answering with only one material, the confidence was 4.19. When creating the tasks, we expected the confidence to drop significantly, but this did not happen, as the confidence levels remained above 4.

4.7.3 User Form

Fig. 4.12 shows a bar plot with the answers given by the users to the first statement of the post-experience form. The first statement was: *"I felt that the different materials improved immersion."* and the possible answers followed a 5-point *Likert scale* [3]. The answers provided by the users were mostly positive, with a total of 85.7% agreeing that using different materials did indeed improve the immersion (38.1% strongly agreed and 47.6% agreed), while 14.3% were neutral. There were no answers that disagreed with the statement. These answerers indicate that the users did indeed find the device able to improve the VR experience.

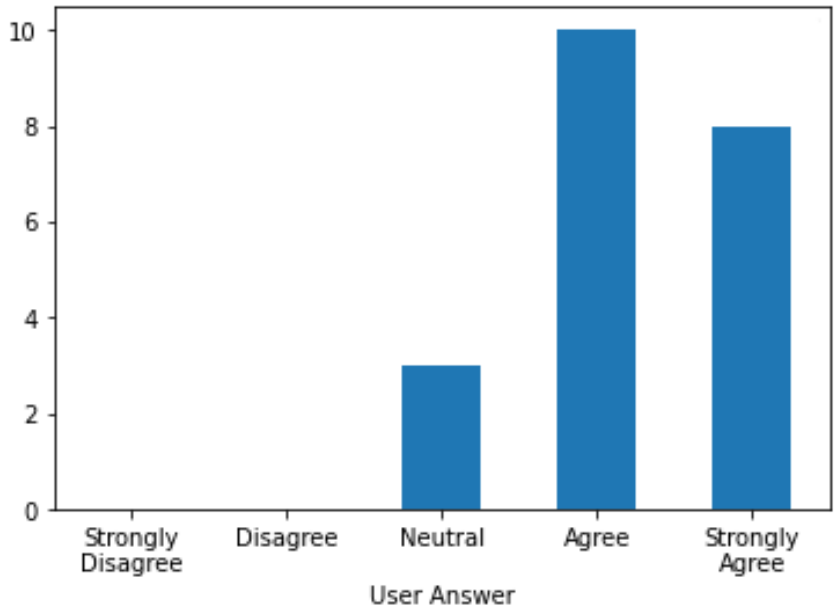

Figure 4.12: Responses to the first statement the users answered: *I felt that the different materials improved immersion.*

Fig. 4.13 shows a bar plot with the answers to the second statement. This statement was: *"I could easily distinguish which material corresponded to the virtual object."*. A total of 71.4% of the users agreed that the materials were distinguishable (9.5% strongly agreed and 61.9% agreed), 23.8% felt neutral, and 4.8% of the users disagreed with the statement. Some users claimed that they did not strongly agree with the statement because they could not feel the material *stone* during the tests. These results are consistent with the results analyzed for the second task, where the material *stone* was selected much less than the other materials. Even though there are fewer

[3] Likert scale (accessed on 15/06/2023): https://en.wikipedia.org/wiki/Likert_scale

Table 4.5: Interquartile range and median of the first two statements.

Statement	IQR	Median
I felt that the different materials improved immersion.	1	Agree
I could easily distinguish which material corresponded to the virtual object.	1	Agree

users that strongly agreed with the statement, there are none that strongly disagreed with it, and only a small percentage disagreed with the statement.

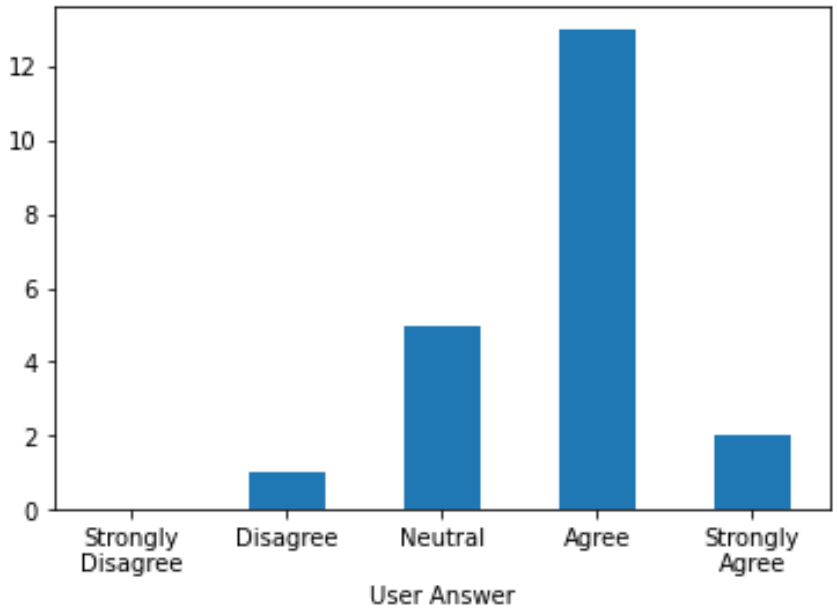

Figure 4.13: Responses to the second statement the users answered: *"I could easily distinguish which material corresponded to the virtual object."*

Table 4.5 shows the median and IQR of the first two statements. Both these statements had the same IQR and median, these being 1 and *"Agree"*(4) respectively. This IQR of 1 shows that the users' answers did not have much dispersion, making their answers concentrate around the *"Agree"*(4) statement.

Finally, Fig. 4.14 shows a bar plot detailing the number of materials people thought the device contained. The most popular answer was 4, with 6 users reporting this value. Some also commented that the materials they predicted were in the device were the main three, these being *metal*, *wood*, and *foam*, but they also predicted *glass* as a fourth material. Out of the 21 users that participated in the experiment, only 4 predicted that the device had only 3 materials, which is a low enough count to consider that there is some potential in using the same physical material to represent different virtual objects.

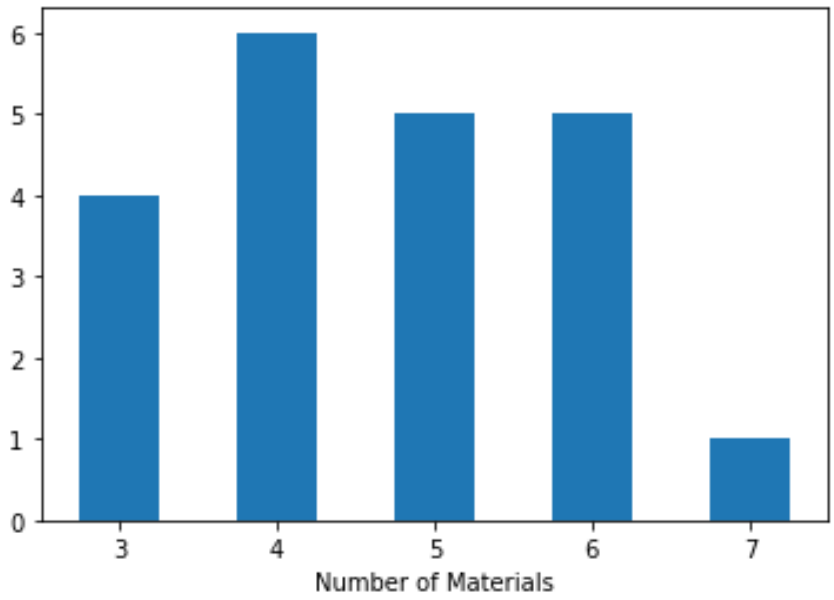

Figure 4.14: Number of materials the users thought the device had.

One thing that hindered the ability of the user to answer this statement is the fact that the users could feel the rough design of the structure even while wearing the HMD. Some users reported that, while grabbing the device, their fingertips would not be supported by the out-facing surface of the prism, but they would be wrapping around it. This wrap of the prism lets the users get a rough estimate of its structure, narrowing down on the answer they provided.

4.8 Summary

This chapter explains how the testing for the solution was done. It details the two tasks presented to the users, and the materials' selection process. It also details the groups done and the order in which these materials are displayed to the user inside the VR application.

The results obtained from the user tests proved to be satisfactory overall. Although there were some materials that could be changed, more specifically the *stone* material, the users were able to correctly distinguish what materials were being simulated. The groups done by the users with the materials also proved to be consistent with the groups made beforehand. The results obtained from the two tasks proved useful for the validation of the device, as well as to verify if the goals set at the beginning of the development were met.

It also details some shortcomings of the structure only detected during the testing phase, more specifically, the users' fingers entering the gap between the prisms. The *Trivision* mechanism proved efficient for the goal of the device, as there were no problems with rotating the prisms synchronously, and correctly change the device's surface to match the correct material.

Chapter 5

Conclusions

The use of virtual reality technologies has been getting more widespread throughout the years. Contrary to the growth of its popularity, little has been done to improve the feedback the controllers provide. The need to physically represent different virtual objects still has a lot of areas where innovation can still be done.

We proposed a device that transform so that its exterior surface can expose different materials to the user interacting with the virtual environment. By doing this, we aimed to increase the users' immersion while they experience virtual reality. We also hoped to create a device that does not hinder the user's movement while being able to change its surface material. This device must also be usable as an input system for virtual reality and be graspable and untethered.

Based on the research, we arrived at a structural design similar to *Shape-a-getti*, while the materials are changed based on the *Trivision* mechanism. A central gear is responsible for rotating every other smaller gear connected to the prisms containing the materials.

We validated the developed solution by performing user evaluations and having the users interact with the device. The solution proved effective in mimicking different materials, and the results also showed that the same physical material can be used to represent different virtual objects.

We believe to have achieved every goal that was planned at the start of this thesis, namely:

- **Design and prototype a solution that allows for the simulation of multiple materials or textures.**
  The *Materializer* proved to be able to simulate multiple materials, as was proven during the user tests, in which the device was able to simulate 7 materials. Although there were only three physical materials, most users felt like there were more than three. The answers to the form also proved that the users agreed that the device improved the experience, confirming the quality of the representations.

- **The solution should be generic so that adding new materials can be done with little effort.**
  By adding a little gap to the prisms, this allowed for easy access to the materials. This made swapping between them very easy and straightforward.

- **The solution should serve as an input system to allow the user to interact with the virtual.**
  Materializer allows for *Bluetooth* communication with the application developed to control it. This, paired with an *HTC Vive* tracker placed on top of the device, allows for a graspable and ungrounded device capable of being used as an input system. On top of that, the interface used for serial communication is generic enough to be used in other contexts other than unity.

5.1 Future Work

The work carried in this thesis leaves room for future research and improvements. Some of that possible future work is outlined next.

- **Different Materials**

  The user tests done with the device only explored three materials. Another study should be done containing different materials from the three already used to further explore the device's capabilities.

- **Different prisms shapes**

  The only prism shape that was explored was a triangular prism. Different prisms with different numbers of sides can also be used to improve the solution's performance. By adding more sides to the prism, more materials can be placed on the device, letting the solution recreate many more materials. One downside of this change is the gap between the prisms grow larger with the more sides the prisms have. The prisms tested only had three sides to minimize the gap, but the device can also work with more sides.

- **More Solid Structure**

  The device used a rubber band to hold the prisms together and force them to be in constant contact with the main gear. This rubber band can be removed by improving the design of the prisms or by changing the design of the base lid. Another design flaw that can be improved is the roof. By removing the possible roof rotation, we would create a more stable design.

- **Add More Prisms**

  Increasing the number of prisms that make up the device makes the gap between them smaller. This smaller gap reduces the problem of the fingertips of the user wrapping around the prism, giving them a rough estimate of the prism's shape. The downside of this solution

is that some pieces used might be harder to create, namely the gears and the material strips. The gears would need to be much smaller, which may make it more difficult to control them to make the prisms rotate. The material strips also need to be much smaller, which requires much more precision when creating these strips and when inserting them into the gaps of the prisms.

- **Independent Motors**

  Adding an independent motor for each prism makes it possible to do material configurations that are not homogeneous. This would allow for additional objects that are not homogeneous to be represented, increasing the possibilities of the device.

Although there is more work that can be done, the device proved effective and the goals of the thesis were reached. We hope this work can provide useful insight for future innovations in this area of haptic feedback for VR.

References

[1] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. Snake charmer: Physically enabling virtual objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, page 218–226, New York, NY, USA, 2016. Association for Computing Machinery.

[2] Filipe Guedes Barbosa. Shape-a-Getti: A haptic device for getting multiple shapes using a simple actuator. Faculdade De Engenharia Da Universidade Do Porto. MSc Thesis, 2022.

[3] Shaoyu Cai and Kening Zhu. Multi-modal transformer-based tactile signal generation for haptic texture simulation of materials in virtual and augmented reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 810–811, 2022.

[4] Texture Can. Cleaning Sponge Texture (Fabric 0028). https://www.texturecan.com/details/168/, Online. accessed on 01/07/2023.

[5] Texture Can. Old Wood Planks (Wood 0048). https://www.texturecan.com/details/467/, Online. accessed on 01/07/2023.

[6] Texture Can. Sound Proof Acoustic Foam Texture (Fabric 0014). https://www.texturecan.com/details/116/, Online. accessed on 01/07/2023.

[7] Inrak Choi, Heather Culbertson, Mark R. Miller, Alex Olwal, and Sean Follmer. Grabity: A wearable haptic interface for simulating weight and grasping in virtual reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, page 119–130, New York, NY, USA, 2017. Association for Computing Machinery.

[8] Eisuke Fujinawa, Shigeo Yoshida, Yuki Koyama, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. Computational design of hand-held vr controllers using haptic shape illusion. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, VRST '17, New York, NY, USA, 2017. Association for Computing Machinery.

[9] Michele Germani, Maura Mengoni, and Margherita Peruzzini. Electro-tactile device for material texture simulation. *International Journal of Advanced Manufacturing Technology*, 68:2185–2203, 10 2013.

[10] Eric J Gonzalez, Eyal Ofek, Mar Gonzalez-Franco, and Mike Sinclair. X-rings: A hand-mounted 360° shape display for grasping in virtual reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, UIST '21, page 732–742, New York, NY, USA, 2021. Association for Computing Machinery.

[11] Sebastian Günther, Julian Rasch, Dominik Schön, Florian Müller, Martin Schmitz, Jan Riemann, Andrii Matviienko, and Max Mühlhäuser. Smooth as steel wool: Effects of visual stimuli on the haptic perception of roughness in virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[12] Elvin Karana, Paul Hekkert, and Prabhu Kandachar. Meanings of materials through sensorial properties and manufacturing processes. *Materials & Design*, 30(7):2778–2784, 2009.

[13] Yuichiro Katsumoto, Satoru Tokuhisa, and Masa Inakage. Ninja track: Design of electronic toy variable in shape and flexibility. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, page 17–24, New York, NY, USA, 2013. Association for Computing Machinery.

[14] John C. McClelland, Robert J. Teather, and Audrey Girouard. Haptobend: Shape-changing passive haptic feedback in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, page 82–90, New York, NY, USA, 2017. Association for Computing Machinery.

[15] OpenGameArt. Ui-sounds by stumpystrust. https://opengameart.org/content/ui-sounds, Online. accessed on 01/07/2023.

[16] Kuniharu Sakurada, Gowrishankar Ganesh, and Wenwei Yu. A novel tactile feedback system with on-line texture decoding and direct-texture-feedback. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12946–12953, 2021.

[17] Jotaro Shigeyama, Takeru Hashimoto, Shigeo Yoshida, Takuji Narumi, Tomohiro Tanikawa, and Michitaka Hirose. Transcalibur: A weight shifting virtual reality controller for 2d shape rendering based on computational perception model. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–11, New York, NY, USA, 2019. Association for Computing Machinery.

[18] Unity Asset Store. 4K PBR Ground Textures by Se34ms. https://assetstore.unity.com/packages/2d/textures-materials/4k-pbr-ground-textures-206354, Online. accessed on 01/07/2023.

[19] Unity Asset Store. Basic Metal Texture Pack by Joseph Steffensen. https://assetstore.unity.com/packages/2d/textures-materials/metals/basic-metal-texture-pack-37402, Online. accessed on 01/07/2023.

[20] Unity Asset Store. Wooden Crates by Animation Arts Creative GmbH. https://assetstore.unity.com/packages/3d/props/wooden-crates-16599, Online. accessed on 01/07/2023.

[21] Yudai Tanaka, Arata Horie, and Xiang 'Anthony' Chen. Dualvib: Simulating haptic sensation of dynamic mass by combining pseudo-force and texture feedback. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*, VRST '20, New York, NY, USA, 2020. Association for Computing Machinery.

[22] Hsin-Ruey Tsai, Ching-Wen Hung, Tzu-Chun Wu, and Bing-Yu Chen. Elastoscillation: 3d multilevel force feedback for damped oscillation on vr controllers. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Tzu-Yun Wei, Hsin-Ruey Tsai, Yu-So Liao, Chieh Tsai, Yi-Shan Chen, Chi Wang, and Bing-Yu Chen. Elastilinks: Force feedback between vr controllers with dynamic points of application of force. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 1023–1034, New York, NY, USA, 2020. Association for Computing Machinery.

[24] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA, 2018. Association for Computing Machinery.

[25] Humphrey Yang, Tate Johnson, Ke Zhong, Dinesh Patel, Gina Olson, Carmel Majidi, Mohammad Islam, and Lining Yao. Recompfig: Designing dynamically reconfigurable kinematic devices using compliant mechanisms and tensioning cables. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[26] Shigeo Yoshida, Yuqian Sun, and Hideaki Kuzuoka. Pocopo: Handheld pin-based shape display for haptic rendering in virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[27] André Zenner and Antonio Krüger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1285–1294, 2017.

[28] Ayça Şentop Dümen, Mine Koyaz, and Yağdır Çeliker Cenger. Unfolding the material: A proposal of a multi-sensory experience oriented material exhibition medium. *Materials & Design*, 219:110740, 2022.

Appendix A

Materializer Version History

1.6

- Added cylinders to the axles to increase robustness

1.5

- Increased width of the prisms by 1.5mm

1.4

- Increased thickness of walls
- Increased size of X appendix

1.3

- Added numbers to the three surfaces of the prism for better debugging

1.2

- Added cavities to slide in the material instead of gluing them

1.1

- Increased base's connection piece width
- Increased base's screw holes width

1.0

- Initial design

Appendix B

Test Forms

This is the form the users had to fill during the test. This form seeks to collect basic information about the users, such as age, gender, occupation, etc. as well as some more in depth data about the user's experience with the device. The answers to the multiple choice questions are also present under each question.

Starting Form

Age

- 16-20
- 21-30
- 31-40
- 41-50
- More than 50

Gender

- Male
- Female
- Prefer not to say
- Other

Have you tried virtual reality before?

- Yes, but don't use it on a regular basis (less than once a month).
- Yes, I use it monthly

- Yes, I use it weekly
- Yes, I use it daily
- No

What is your current field of study or area of expertise?

Post Test Form

Table B.1: Post Test Statements

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I felt that the different materials improved immersion.	○	○	○	○	○
I could easily distinguish which material corresponded to the virtual object.	○	○	○	○	○

I felt approximately this many different materials:

Additional comments

Appendix C

Microcontroller Code

This is the code used to control the microcontroller. The code is burnt into the microcontroller using *PlatformIO*.

Listing C.1: Microcontroller Code

```
#include <M5Atom.h>
#include <string>
#include <ESP32Servo.h>

#include "DevState.hpp"

#include "BluetoothSerial.h"
BluetoothSerial SerialBT;


#define MAIN_MOTOR_PIN 25

Servo mainServo;

bool stringComplete = false;
bool rotationCommand = false;
String inputString;
int lastValue;


void BTcallback(esp_spp_cb_event_t event, esp_spp_cb_param_t *param){
 //Serial.println(event);
 if(event == ESP_SPP_SRV_OPEN_EVT){
   Serial.println("BT Client Connected");
   setState(DEV_NORMAL);
 }
 if(event == ESP_SPP_SRV_STOP_EVT || event == ESP_SPP_CLOSE_EVT){
   Serial.println("BT Client Disconnected");
```

```
   setState(DEV_CONNECTING);
 }

}

void setup() {
  Serial.begin(115200);
  M5.begin();

  setState(DEV_STARTING);

  SerialBT.begin("Material Changer");
  SerialBT.register_callback(BTcallback);

  mainServo.attach(MAIN_MOTOR_PIN);
  mainServo.write(0);
  lastValue = 0;
}

void loop() {
  M5.update(); //You need to add m5.update () to read the state of the
      key, see System for details
  if (M5.Btn.isPressed()) { //If the button is pressed
    Serial.println("Button is pressed.");
  }
  if(stringComplete){
    if(rotationCommand){
      Serial.println("Rotating to " + inputString);
      mainServo.write(inputString.toInt());
    }
    stringComplete = false;
    inputString = "";
  }
  delay(20);

  while (SerialBT.available()) {
    // get the new byte:
    char inChar = (char)SerialBT.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if(inputString.equals("R=")){
      rotationCommand = true;
      inputString = "";
```

```
    }
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}

void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if(inputString.equals("R=")){
      rotationCommand = true;
      inputString = "";
    }
    if (inChar == '\n') {
      stringComplete = true;
    }
  }

}
```