Aalborg University Copenhagen

Semester: 10th

Title: Psychophysical experiment examining object properties of physical props in virtual reality

Project Period: 1st February 2021 – 28th May 2021

Aalborg University Copenhagen

Frederikskaj 12,

DK-2450 Copenhagen SV

Semester Coordinator: Stefania Serafin
Secretary: Lisbeth Nykjær

Semester Theme:

Master thesis

Supervisor(s):

Niels Christian Nillson

Project group no.:

Members:
Márk Szekér
Nicky Robert Toma

Abstract:

This project researched haptics and object physical properties to ascertain whether some specific properties are more important than other ones, and what level of mismatch is tolerable between a physical prop and a virtual object representing the prop. Two objects have been chosen, a pan and a hammer to test the mismatch between the versions. The chosen properties were surface texture, size, and shape. Different papers have been analyzed which include techniques that can be used to perform the experiment. The two objects were modelled and textured to resemble the real-world objects as best as possible, and their surface texture, size and shape were given three levels of fidelity, with 0 being the most inaccurate. An evaluation was performed at Aalborg University Copenhagen campus, but due to the COVID pandemic the sample size was very small. Regardless, the data was analyzed and the results show that size is more important than surface texture and shape when matching virtual props with physical objects. The data also showed that surface texture and shape could be at fidelity level 1 to register a match between the virtual object and physical prop.

AALBORG UNIVERSITY

Psychophysical experiment examining object properties of physical props in virtual reality

Authors:

Márk SZEKÉR

Nicky Robert TOMA

Medialogy, 10th semester

Supervisor:

Niels C. NILSSON

May 28, 2021

AALBORG UNIVERSITY

STUDENT REPORT

Contents

Chapter 1

Introduction

As technology advances, virtual reality (VR) experiences are becoming more and more realistic and immersive. Higher and higher fidelity audio-visual experiences are provided by the rapidly developing head mounted display (HMD) systems. However, while these systems stimulate our vision and hearing, our other senses remain neglected. The use of passive haptics, in the form of physical props, is one way researchers have been trying to enhance these VR experiences, while stimulating the haptic perception of people [1] [2] [3] [4] [5]. However, while the use of props may enhance the user experience, it also introduces an added level of complexity to the system and poses new challenges to designers and developers. As the virtual environments (VE) become more and more complex, it becomes increasingly difficult to map every interactable object onto a physical prop [6]. Several studies looked at how props could be reused within the same VE, as well as dynamic props, that can change some of their haptic properties in real time [7] [2] [8]. In order to simplify prop design and procurement, the use of everyday objects as physical proxies for a VE has also been researched [1]. Some of these methods inherently include some level of mismatch between the prop and the virtual object they are mapped to [9]. While it has been investigated to what degree do users tolerate this mismatch [10] [9], the research in the field is far from comprehensive.

Several studies looked at how mismatches in the VE influence the users' sense of presence or embodiment. Skarbez et al. [11] and Slater et al. [12] used a technique in which users were shown a high fidelity VE, which they later had to try to recreate, starting from a low fidelity version of the same scene and adjusting certain properties. The study analyzed how close to the original users had to go with the level of detail for it to feel the same, as well as the order in which they adjusted the different properties. In this study we aim to take a similar look at physical props. Our aim is to find out what physical properties users consider important, when using a physical prop in VR and what level of mismatch they accept between the prop, and the virtual object it is used to represent. Our findings may have implications on future prop design, as well as the incorporation of already existing objects in a VE, or the repeated use of the same object to represent multiple virtual objects. Our work was inspired by the following initial problem statement:

What level of mismatch is tolerated between a physical prop and virtual item for them to still be considered the same object, and what object haptic properties are most important when matching the virtual object to its physical counterpart?

Chapter 2

Analysis

This section will detail haptic feedback and the types of haptics that are present in virtual reality. It will introduce material and geometric properties of objects, in order to ascertain which could be analyzed in this project. It will also detail techniques that could be used in quantifying physical properties. Finally the section will end with the formation of a final problem statement.

2.1 Haptics

Haptics is believed to be a word derived from the Greek word *haptesthai* which means *related to the sense of touch* [13]. The field of haptics has been studied in more than one professional field. For example in physiology it means the study of human touch through kinesthetic (force/position) and cutaneous (tactile) receptors [13]. In virtual reality the study of haptics focuses on the development of interfaces that allow the users to experience the sense of touch in a virtual environment [13]. Haptic sense is extremely important in day to day life as it allows us to interact and identify the properties of objects (texture, hardness, temperature, shape, size, weight) [14][15].

2.1.1 Human haptic perception

Haptic sensing differs from other human senses like vision or hearing in terms of its localization. Vision and hearing are localized to a specific body region, while the haptic sense is distributed across the entire body [15], through skin, muscles and tendons. As mentioned previously, haptic sensing is split into two modalities: kinesthetic and cutaneous. Kinesthetic sensations consist of forces and torques and are sensed in muscle and joints, while cutaneous sensations (pressure, shear, vibration) are sensed in the skin, particularly in mechanoreceptors [15]. There are four types of mechanoreceptors, and they differ in terms of signals captured:

- **Meissner corpuscles** - fast adapting receptors that respond to low frequency vibrations (5-50 Hz) and skin deformation [15][16]
- **Pacinian corpuscles** - provide information about transient contact that respond to a wider range of frequencies (40-400 Hz) [15][16]
- **Merkel disks** - slow adapting receptors that identify edges and spatial features [15][16]
- **Ruffini endings** - receptors that sense skin stretch and direction of objects [15][16]

These mechanoreceptors are not evenly distributed across the entire body. They are more numerous in the glabrous skin of hands and feet compared to hairy skin [15]. Mechanoreceptors allow humans to distinguish object haptic properties consisting of material(texture, hardness, temperature, weight) and geometric(weight, shape, size) properties [14]. According to Lederman and Klatzky [14] material properties are easier to distinguish by using the haptic sense while geometric properties are more easily distinguishable by using the visual sense.

2.2 Types of haptic feedback in VR

Haptic feedback in VR can be split into more categories based on whether the feedback is coming from computer controlled actuators or the presence of the tool itself [17]. These categories are:

- **Active haptics:** Active haptics represent techniques that use computer controlled actuators to provide the feedback [17]. Examples include graspable systems that are typically kinesthetic (force-feedback) devices that are grounded and allow the users to push on them [15]. They can also be wearable systems (cutaneous) that are mounted to the hands or other parts of the body and apply sensations directly to the skin [15]. They can provide vibration, lateral skin stretch, and normal skin deformation. Finally touchable systems, that can be purely cutaneous devices and change their tactile properties based on location. They can also be hybrid cutaneous and kinesthetic devices that change their shape, mechanical properties and surface properties

- **Passive haptics:** Passive haptics refers to techniques where the haptic feedback comes from the use of the physical prop itself. The use of physical props is one of the most common practice of using passive haptics to provide feedback. An advantage of passive haptics is that it is cheaper to provide [18]. Passive haptics have been used in redirected walking [19] [20] and haptic retargeting [7] [21]. One example of a project that does not use physical props to provide the haptic feedback, but instead uses an elastic band to connect the user's hand to their knees and thus provide resistance feedback, is the Elastic-Arm [22].

- **Dynamic-passive haptics:** Dynamic-passive haptics refers to technology that uses a mix of active and passive haptics to provide feedback. It was coined by André Zenner and Antonio Krüger when they developed Shifty [2]. Shifty works by changing it's center of gravity using weights mounted inside the haptic prop and this way can provide representations of more objects in VR. Another example of a dynamic-passive haptic system is DragOn [8], again developed by André Zenner and Antonio Krüger. This product provides dynamic passive haptic feedback based on drag and weight shift.

Figure 2.1 shows the active and passive haptic continuum with dynamic passive haptics somewhere in between. This project will make use of passive haptics, in the shape of physical props, due to the difficulties related to construction involved in working with active haptics, and their price.

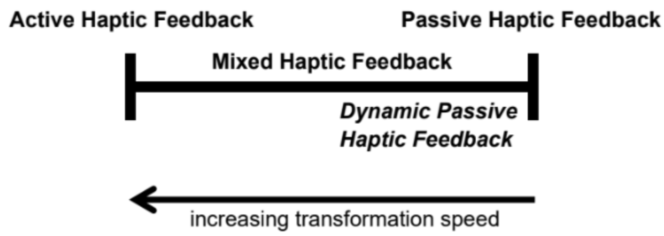


Figure 2.1: Active passive continuum.

2.3 Physical props in virtual reality

Using physical props to represent virtual objects offers a convenient way of providing haptic feedback to the users of VR. However, one must take certain things into account when designing such props, in order for the interactions to be believable. Nilsson et al. [6] proposed a set of success criteria for designing such props:

- **Sufficient similarity:** Every prop that is manipulated by the user should be sufficiently similar to their virtual representation in their haptic properties.
- **Complete co-location:** The virtual objects the user touches should be perfectly co-located with a physical prop.
- **Compelling contact forces:** If the user touches a virtual object that is affected by some force in the VE, compelling stimuli should be presented to the user.

The criteria of sufficient similarity and complete co-location can be satisfied independently of each other, however when it comes to compelling contact forces the fulfillment of this criterion depends on the fulfillment of the other two. For example, if every object in a VE is mapped to a physical prop, then all interaction between a grasped object and the VE will deliver compelling contact forces as a result of the physical props interacting with each other. Both Nilsson et al. [6] and Simeone et al. [9] argue that the degree to which these criteria should be met depends on the application of the VE. In [6] it is also mentioned that while there is an increasing amount of research dealing with the amount of mismatch the users will tolerate between the virtual and the physical objects, these effects are not fully understood.

2.4 Quantifying haptic perception of virtual objects

This section will discuss several projects that use various research techniques that could be used to quantify haptic properties of objects in a virtual environment. Projects developed by Skarbez et al [11] and Slater et al. [12] that use psychophysical methods to determine what factors are more important for plausibility (PSI) and place (PI) illusion in VR will be detailed. The same technique could be used in this project to figure out what haptic properties are more important to get right when dealing with physical props and their VR counterpart (eg. is the shape of a virtual object more important than its texture, to believe that it represents the physical prop that one is holding). Another project from Bergström et al. [23] will be discussed as it deals with an illusion that can change the perception of size in VR. Finally we will detail work done by Simenone et al. [9] that investigates different haptic properties of objects in VR and how they tie in with believably of haptic props.

2.4.1 Psychophysical methods

Skarbez et al [11] recreated a VE consisting of a bar, with different non-playable characters (NPC's), and a mirror in front of the user in order for them to be able to see their avatars in the scene. Their goal was to investigate coherence factors [24] and how they relate to PSI and PI in VR. The experimenters varied four coherence factors: Virtual human behaviour coherence (VH), virtual body behaviour coherence (VB), physical coherence (P) and scenario coherence (S). Each of these factors had different levels of fidelity, with 0 being the lowest level and 2 the highest, amounting to 81 different possible configurations.

The experiment they performed consisted of the users interacting with the system for the first time at maximum level of all factors. They had to experience the scene, move around, kick a football and take note of how they feel in terms of PSI and PI, in this maximum level of the factors. After experiencing the scene for the first time participants were exposed to one of 8 random conditions that had lower levels of coherence factors, for example all factors at level 0 or one factor at level 1 and the rest at 0. During the course of the experiment the participants were instructed to incrementally change any one of the coherence factors, until they get to a level of PSI or PI that was observed during their initial interaction with the system. The average number of improvements the participants needed to get to the same level of coherence as in the first interaction was 6.9.

The researchers came to the following conclusions after performing the experiment: the most important factor of PSI is the virtual body, the other factors of coherence can remain in level 1 not necessarily 2. The second most important factor was scenario coherence. The researchers also speculate that users that played with the ball more interacted more with the scenario so they felt a higher degree of presence. Because the ball was more important to them they did not accept configurations where the ball was behaving unrealistically. As a last find they also argue that VB is a factor of scenario that affects PSI, while the interface that controls the virtual body is a factor that affects PI.

Another project that studies presence, PSI and PI is the project developed by Slater et al. [12]. Similar to the project by Skarbez et al. [11] they had 4 properties that they controlled, to manipulate PSI and PI: illumination (I), consisting of 3 levels, field of view (F), consisting of 2 levels (D), display type again consisting of 2 levels, and finally the virtual body (V), consisting of 3 levels. The first time participants experienced the scene, the properties were at their highest levels, and they were allowed to modify each of them to see the effect. While experiencing the scene, half of them were instructed to pay attention to PSI and half to PI. After this training period, participants performed 5 trials starting from a basic configuration of properties, and were told to modify the configuration and stop whenever they feel the same level of PSI or PI they felt in the initial session. The researchers imposed some rules so that the participants think carefully about their choices while also making sure they cannot jump to the highest possible configuration in one move: the transitions could be only increased and never decreased, and only one step transitions could be made.

To analyze the data the researchers looked at probability distributions and transition matrices. Their research yielded some findings such as that *"natural sensorimotor contingencies are important for PI"*, with 88% of people in the PI group stopping in a configuration where F and D are at their maximum level. Researchers also got to the conclusion that *"correlations between self-actions and events are important for Psi"* and that *"illumination realism may be more important for Psi"*. The PI group first adjusted the level of F and D and only after, changed the level of the virtual body. Those in the Psi group tended to improve the illumination more. Finally according to the findings, *"the virtual body is important for both Psi and PI"* [12].

2.4.2 Resized grasping

Bergström et al. [23] proposed a technique called resized grasping to represent more virtual objects differing in size with the same physical prop. Resized grasping works by introducing pseudo haptic feedback [25] to "decouple the fingers' virtual and physical locations", for example the user's fingers in physical space are close together and grasping a small object, but in VR space by introducing offsets in the position of the fingers, the user's fingers will be far apart and grasping a large object.

The purpose of the study was to find the threshold between physical and virtual object sizes, to offer developers the possibility of using one physical prop to denote objects of different sizes in VR. For their experiment they used a cube with 3 physical widths of 3,6 and 9 cm respectively and 11 virtual widths from 1 to 11 cm, and they used a two-alternative forced choice to record the data. The task the participants had to perform was very straightforward, all they had to do was pickup a physical cube and put it down again, after which a question would pop up in virtual reality letting them choose whether the virtual object feels smaller or larger than the physical one. The participants were not allowed to see any of the physical cubes.

The results from their experiment suggest that objects can indeed be scaled up and down in VR using illusions of this kind. Smaller objects can represent more objects virtually compared to large objects.

2.4.3 Object substitution

In a 2015 paper Simeone et al. [9] looked at the idea of using physical proxies that have some degree of mismatch compared to their virtual counterpart. They wanted to investigate how this mismatch affects the users' suspension of disbelief, and the believability of the proxies. They designed two studies, that would approach the issue from two different sides.

In the first study, users were placed in a VE representing a medieval courtyard. Each object in the real room, where the study took place, was substituted with a virtual object, to make a compelling VE. The users were asked to manipulate a single object, that was placed in front of them, on top of a barrel. The real object was a mug in each case. However, the virtual object was changed each time, to represent a different level of mismatch from the real object.

They found that the biggest contributing factors to a decreased level of suspension of disbelief were the difference in shape or perceived temperature of the object. They also found significant differences in the similarity perceived by the users when the material or the size of the object was mismatched.

In their second study, they placed users in a VE representing the bridge of a spaceship. The environment was constructed in a similar manner to the first one, in that every object in the real room where the study took place was substituted with a virtual object. The users' task here was to hit a blue ball with a virtual lightsaber, that was represented by a different physical prop in each iteration. The three props used were a replica of a lightsaber, an umbrella and a flashlight.

Their results show that the majority of users preferred the flashlight over the other two objects, mostly because of its weight. Even though the flashlight weighs less than the lightsaber would, their affordances are quite similar in the places that the users were able to touch. The smaller weight compared to the other props used, also resulted in less fatigue, which contributed to the flashlight being the preferred object. The similarity in affordances and the reduced fatigue were enough of a factor to counteract the mismatches between the flashlight and the lightsaber in terms of their other properties.

They conclude that object shape is a major factor when deciding what virtual object to substitute a real one with. However, as long as the functionality is similar, there is a lot of room for alterations. Similarly, emphasis should be placed on matching the shape and affordances of the manipulable parts of the objects, while a larger amount of mismatch is tolerated on other parts.

2.4.4 Conclusion

It must be mentioned that this project will not focus on researching presence. The aforementioned papers [11] [12] are presented purely to illustrate the techniques that the researchers use to collect and analyze the data, as the same techniques can be used for this project. Resized grasping [23] is a way to change perception of size so it could be useful when analyzing size. The paper written by Simeone et al. [9] is a very good example of how to analyze different physical properties when dealing with props and their virtual counterpart.

2.5 Delimitation and final problem statement

As mentioned in section 1 the focus of this project is to analyze what level of mismatch is tolerated between virtual objects and physical props. For this purpose the techniques presented in the papers written by Skarbez et al. [11] and Slater et al. [12] will be used to record and analyze the data. The physical properties chosen for analysis will have different levels and the user will need to match a low quality version of the object with the maximum quality one, by changing the properties incrementally. Unfortunately not all physical properties can be analyzed. As mentioned in section 2.1, material properties (texture, hardness, temperature, weight) are analyzed mostly by interacting with the object and feeling it, so unless there are modifications to the physical prop it is difficult to analyze them. Geometric properties (weight, size, shape) are analyzed by using the visual sense, so for this project, the properties that will be analyzed will be mostly geometric, namely size and shape. Surface texture can be analyzed by adding bump maps to the objects, so it will also be included. Following this research, the following final problem statement has been constructed:

Which of the following physical properties: surface texture, size, shape is most important when trying to match physical props with their virtual counterpart and what level of mismatch is tolerable?

Chapter 3

Research methodology

This section of the report will detail where and how the experiment took place, and also expand on the ways the data was recorded and analyzed. The purpose of this experiment was to ascertain which of the following three physical properties: surface texture, size, and shape, are most important to participants, when handling a physical prop, and a virtual object representing that prop in VR. More specifically, given inferior levels of each of the factors, how much and in what order would they increase these factors to achieve a feeling that they are indeed handling the same object in physical space that they are looking at in the virtual environment.

3.1 Location and equipment

The experiment took place at Aalborg University Copenhagen in an isolated room to minimize any confounding variables that could impact the data. The equipment used for this experiment was an Oculus Quest 2, a laptop that is connected to the Quest, and two objects, a pan and a hammer, which were tracked by using the Oculus controllers. One reason for using these objects was because there was no access to the university campus this semester due to the COVID pandemic, so we had to use objects that we had at home, which also included an easy point of attachment for the controllers, in this case the handle of the hammer and the pan. These two objects also allow for the testing of the surface texture property, as they have grainy textures or engravings on them.

3.2 Data recording and experimental procedures

For this purpose we used techniques similar to what Skarbez et al. [11] and Slater et al. [12] used when they investigated how different factors with different levels affect place and plausibility illusion in VR. As mentioned earlier there were three material properties investigated, which were surface texture, size, and shape, each with three different levels. Each instance of the three factors is called a configuration and it is denoted by a vector Similarity = {Surface texture, Size, Shape}. Surface texture is tied to the bump map in the objects' texture file. This means having no bump map at the lowest level, after which the strength of the bump map is increased in each configuration. According to [9] there is a higher mismatch between virtual and physical objects if the virtual object is smaller than the physical one, compared to the mismatch when the virtual object is bigger. Because of this it was decided to have the virtual object be incrementally smaller, and not bigger than the real size in any case. The shape attribute is related to the polygon count of the objects, which is gradually increased in each configuration. The different levels of the properties are presented in detail in section 5.3.2. Having three properties with three different level amounts to a total of 27 configurations. See figure 3.1 for all possible configurations.

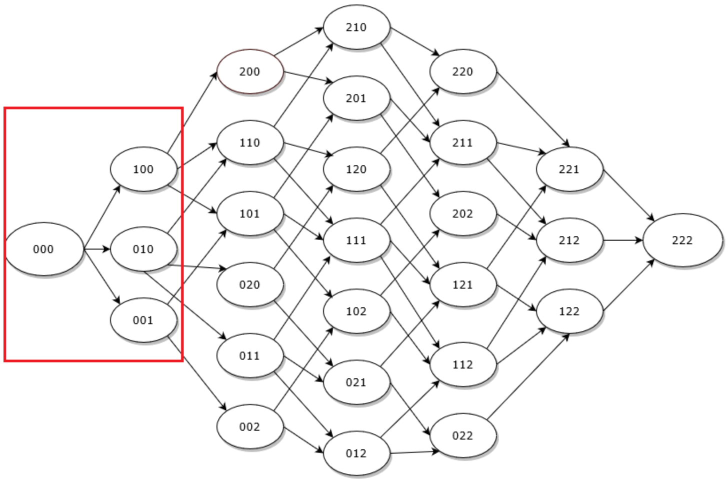


Figure 3.1: All possible configurations of the three properties. The states denote the vector Similarity = {Surface texture, Size, Shape} .

One reason for choosing these physical properties was that they could be manipulated without altering the prop in some way. Another reason was that according to Simeone et al. [9] mismatches were found between the objects when size, shape and materials were different, so these three were meaningful to test to see if there are differences between their importance. As mentioned in section 2.1.1 material properties are recognized by touching the object while geometrical properties are recognized by looking at it. Because, unfortunately there was no access to the campus this semester, it was not possible to create different physical props with different material properties. The only material property that was viable to test was "surface texture" because it could be controlled by using bump maps, and different levels of bumps maps.

The independent variables in this experiment are the three different properties and different levels, while the dependent variable is the sense of similarity that the participants feel between the virtual and physical prop based on these three different factors and levels. The experiment proceeded in the following way: at first participants had to read and sign a consent form, after which they completed a demographic survey. Next, they were instructed on how the test will proceed and what their tasks are, and finally, after the experiment they completed a questionnaire where they had to rate the importance of each factor on a 5 point Likert scale, and argue for why they did so.

The task the participants had to perform during the experiment was to get familiar with an object in physical space by touching and feeling it while looking at a high detail replica of the object in virtual reality. After this, they were presented with a lower detail version of the same object, and they had to increase the properties until they felt that it matches the initial version. To get familiar with the objects, the participants had a time window (minimum 5 seconds, but they could use more) at the start of the test to get the feel of the props. Then they were instructed to remember this state and try to match it (this is called ideal configuration), after which they were switched to a case where all the properties are level zero or one of them is level one with the rest being zero. Observe figure 3.1, the red box represents the initial four configurations that participants started the experiment in. The participants had to complete a trial for each configuration in a random order, amounting to four trials per object, a total of eight per participant. The participants were only able to increase the level of one property at a time, had to increase them incrementally and they were not able to go back once they increased a level. It needs to be mentioned that the initial state that the participants had to match, was the configuration denoted Similarity = {222}, but participants were told that it is a random configuration so that they would not increase all the properties to the maximum on account of knowing that this was the initial version they saw. The data recorded from the experiment was organized into a diagram as in the previously mentioned papers [11] [12], and in addition to that there are also the responses from the survey that were analyzed using statistical methods. The diagram was analyzed to see which property the participants deemed as being most important in this experiment, and it also allowed for the calculation of what the probability is that a given configuration is accepted if it was reached. In addition to this it also allowed us to analyze the different paths the participants took to reach the ideal configuration.

Chapter 4

Design

This section will detail the design of the application that was developed to investigate the problem.

4.1 Hardware

Due to the coronavirus pandemic, access to the school campus was still limited during the semester, thus we had to impose a lot of limitations on what we could use during design and development. The initial idea was to use an HTC Vive[1], with its dedicated trackers being used to track the physical props in the experiment. However, due to the current situation, we had to change the platform we were using. The school was able to provide us with an Oculus Quest 2[2], so this meant that tracking had to be facilitated using the controllers of the Quest.

Furthermore, our initial idea was to manufacture props for use during the experiment, which would then also be modeled in a 3D modeling software. This would have allowed us great flexibility in terms of our design choices, but in the end this was not possible either. Instead, we had to use household items as the props, and make 3D models of those. The choice fell on a frying pan and a hammer, mostly because these were the items we have found to be relatively easy to model in the time available, and allow for the controllers to be attached using non-destructive methods.

[1] https://www.vive.com/eu/
[2] https://www.oculus.com/quest-2/

Figure 4.1: The frying pan and the hammer, with the Oculus controllers attached to them.

4.2 Software

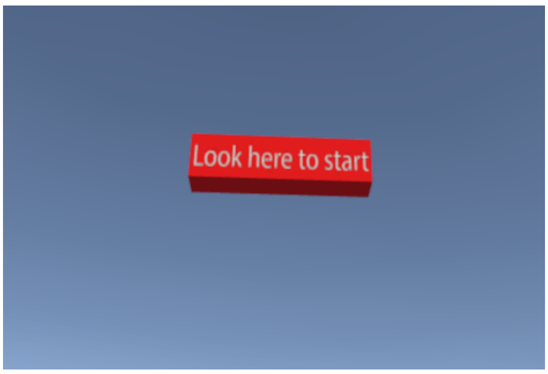
Similar things are applicable in terms of the software component, as were for the hardware. The ongoing pandemic forced us to make significant changes to our initial ideas. For example, the users' interaction with the objects was severely limited. Initially, we planned to design a few tasks, that would require the users to interact with the objects, and use them in some way that would be organic to a VR scenario, as well as the objects themselves. This was reduced to the users inspecting the objects and trying to decide how similar the virtual representations are to the real objects.

At first we also planned to have the users "alter" the physical properties of the physical props. However, since this would have required us to manufacture a large number of props, with slight alterations, we have decided to move the variable properties onto the software side.

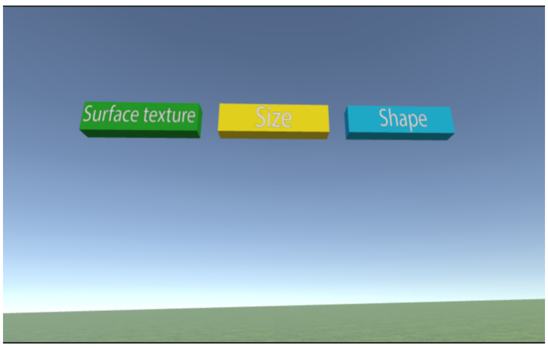
Thus, in the implemented software the user is able to alter the visually identifiable physical properties of the virtual object. These are **surface texture, size** and **shape**. How these properties are represented in the software is discussed in detail in section 5.3.2. Upon launching the application the user is first presented with a version of the virtual object that closely matches the physical prop in terms of these properties.

Figure 4.2: The virtual hammer, with the controller also visible.

They then have a minimum of 5 seconds to inspect the physical object with their hands, while looking at the virtual object. After the time expires, a start button appears inside the VE (see figure 4.3), but the user may continue to inspect the object as they see fit. When they are ready to begin, they have to click the start button. At this point, the virtual object is replaced with a version that significantly differs from the physical object in terms of the aforementioned properties. Then, the user has another 5 seconds (or more if they need it) to inspect the object, after which three buttons become visible inside the VE (see figure 4.3), which the users can use to enhance one of these properties. The buttons can be activated by looking at them for 3 seconds. These cycles (referred to as cases) repeat as long as the user hasn't reached the version of the virtual object, that is closest matching to the physical prop. At this point the user has to signal that they have reached similarity, and the application will be restarted.



(a) The start button



(b) The buttons used to adjust the properties of the virtual objects

Figure 4.3: The different buttons used in the application

Chapter 5

Implementation

This chapter will explain how the application was implemented. First the 3D models will be described that were created for use in the application. Details about the modeling and texturing process will be disclosed. Following this, the application developed to run on the Oculus Quest 2 will be described, complete with screenshots and code examples.

5.1 3D models

This section will detail the software applications and procedures used when modelling and texturing the 3D models of a pan and hammer that were used in testing

5.1.1 Frying pan

The frying pan was modeled in Blender [1], which is a free tool that allows for creation and texturing of 3D models. The pan that served as a reference for the virtual object can be seen in figure 5.1 When starting to model the pan, it was decided to have it split into more parts so that modifications needed in certain areas would be easier to accomplish. The pan was split into the body, handle and the piece of metal connecting the aforementioned parts.

Figure 5.1: Physical prop used as reference for the 3D model.

To model the body a circle was used which was then filled out and modelled accordingly using reference picture, to get as similar as possible to the physical prop. To get to the correct shape operations such as extrude and inset were used, in addition to moving around vertices and edges.

[1]https://www.blender.org/

The most difficult part was the "lip" of the pan, that required extruding, inserting and careful movement of the edges to get right. The low polygon version of the container can be seen in figure 5.2.

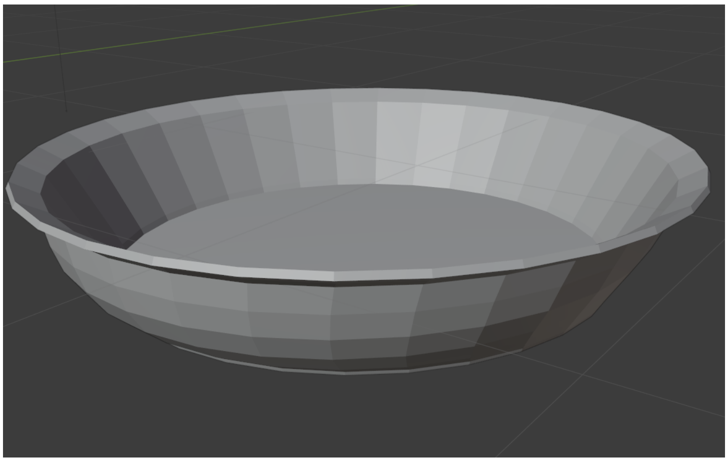
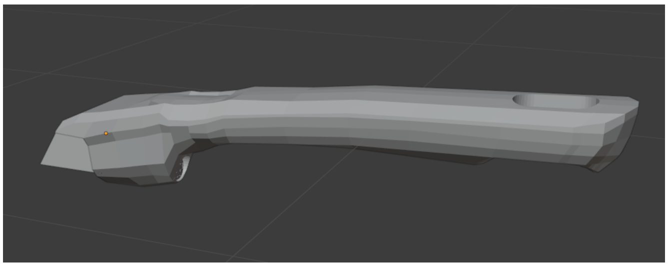
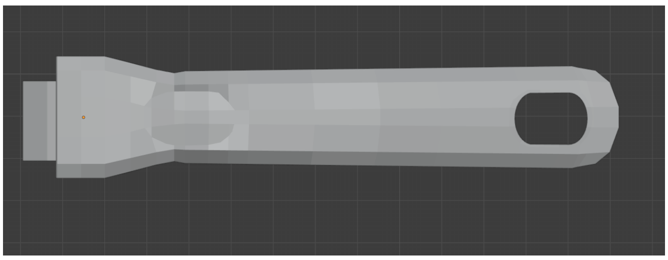


Figure 5.2: Low polygon version of the pan body

The connecting part between the body and the handle was the simplest to add, it being only a cube that had one of its faces modified, so it will be included while presenting the handle. The low polygon version of the handle can be seen in 5.3. The handle constituted the most work when considering the pan. The difficulty arose due to the many curves that the handle incorporates in its design. To model the handle so that it matches the physical prop, pictures were taken of the handle from different perspectives to serve as a reference. These pictures can be seen in figure 5.4.



(a) Low polygon version of the handle and connector. Side view



(b) Low polygon version of the handle and connector. Top view

Figure 5.3: Low polygon version of the pan handle

(a) Reference picture showing the handle from the right

(b) Reference picture showing the handle from the top

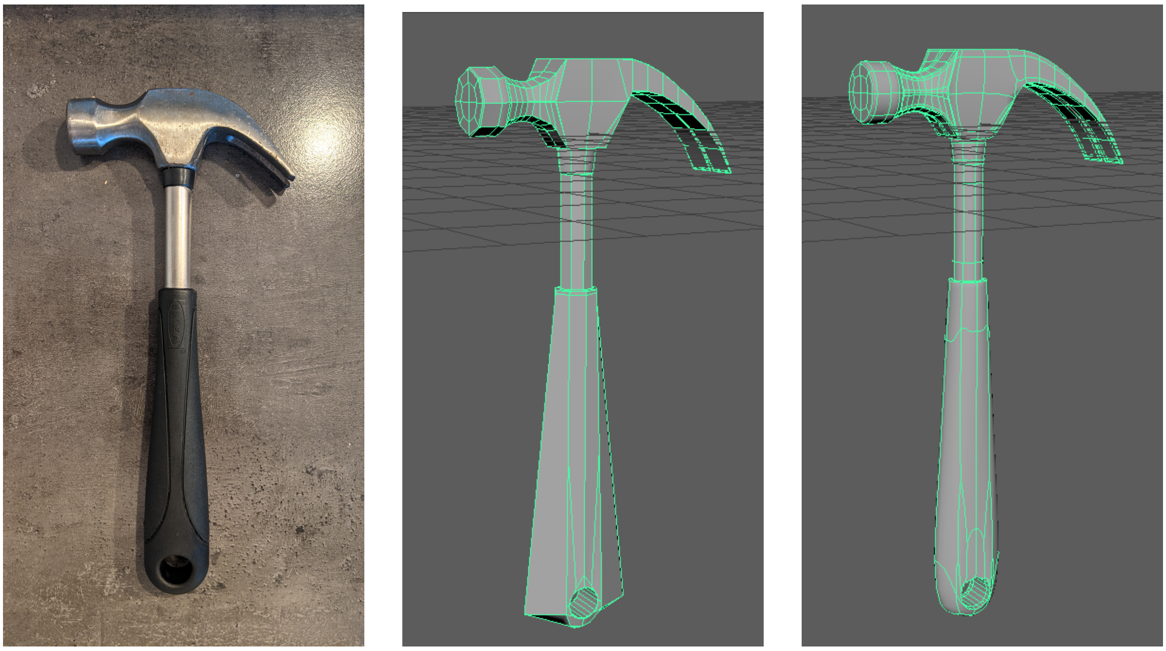
Figure 5.4: Reference pictures used for modelling the handle

The handle started off as a cube and had to be adjusted quite a lot to get it into the desired shape. This included mostly extruding parts of the cube, modifying the edges and adding loop cuts on the model to be able to get more complex shapes. To make this process easier it was decided to work on one half of the handle and use the mirror tool to create the other half, so that time would not be spent on making the two sides identical. As it can be seen in figure 5.4 the handle has a hole in it. To make the hole in the virtual object the boolean tool was used which allows for cutting and combining objects using other objects. In this case a smaller shape that matches the hole was made using a cylinder, which was then used to cut a hole in the handle. This concludes the modelling of the pan. To sum it up it was a process involving mostly adding edge loops and modifying vertices to match the shape of the handle, with operations like boolean and extrude being used in certain situations.

5.1.2 Hammer

The hammer model was made using Autodesk Maya[2]. It started out as a Maya default cube object. Most of the features were created by extruding parts of the cube, while also adding edge loops when necessary. The vertices and edges were moved around in order to get the desired shapes. A reference object was used to create a model as lifelike as possible. The hole on the handle was made using the *boolean* function, as described in section 5.1.1.

[2]https://www.autodesk.eu/products/maya/overview



(a) The reference item

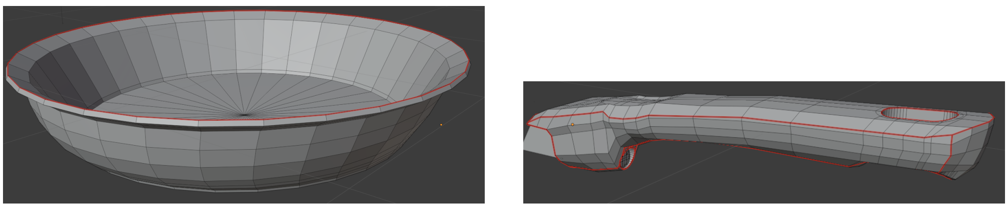
(b) The hammer model, with low polygon count

(c) The hammer model, with high polygon count

Figure 5.5: The hammer prop, and the model with different polygon counts

5.2 Texturing

This section will present the process of texturing the objects, both in terms of general texture and also bump maps. The textures were applied using Blender. The process consisted of marking seams on the objects in order to unwrap a UV map for the textures. The seams for both major parts of the pan can be seen in figure 5.6.

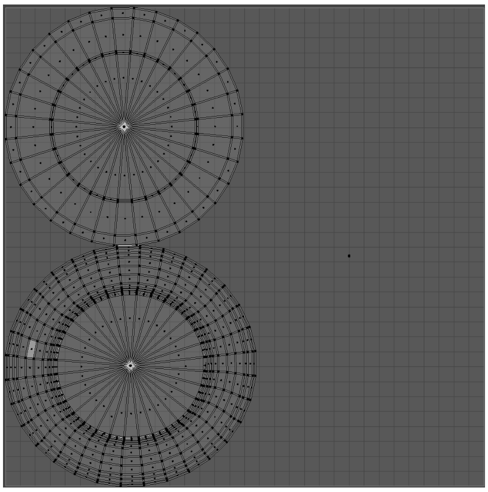


(a) Pan container seams.

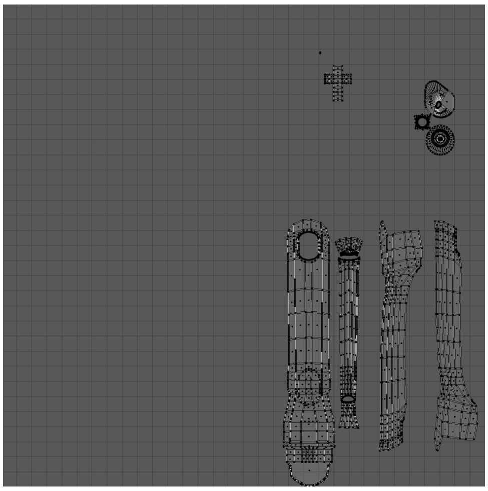
(b) Pan handle seams.

Figure 5.6: Seams

The next step after marking the seams on the objects is to unwrap the UV maps to prepare them for exporting. Basically how UV mapping works is that the maps get exported and opened in an image editor, and the textures are arranged on the maps accordingly. The UV maps for the major parts of the pan can be seeen in figure 5.7. The next step consists of finding textures for the objects. All textures used for the objects are copyright free and they consist of seamless pictures of materials, for example steel and teflon for the pan. The textures that were used for the pan can be seen in figure 5.8.

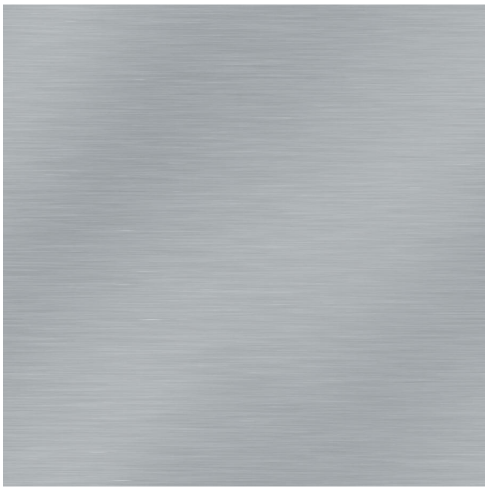


(a) Pan container UV map.



(b) Pan handle UV map.

Figure 5.7: Pan UV maps



(a) Steel texture.

(b) Teflon texture.

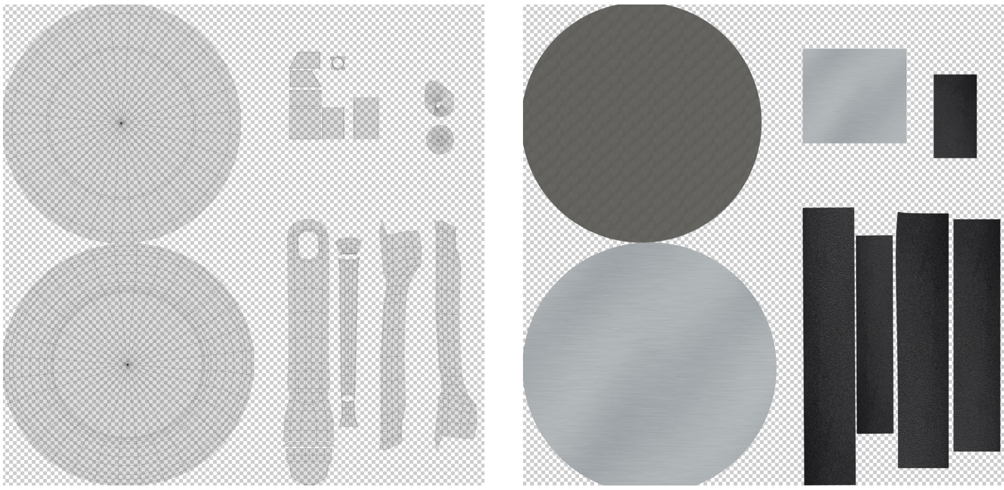
(c) Handle plastic texture.

Figure 5.8: Pan textures

The next step in the texturing process consists of using an image editing software to arrange the textures on the UV map. For this purpose Adobe Photohop CC 2019[3] was used. The combined UV map for entire pan can be seen in figure 5.9. To apply the textures to the UV map simple

[3]https://www.adobe.com/dk/products/photoshop.html

operations of pasting were used. The textures applied to the UV map can be seen in figure 5.9, second image.



(a) Pan UV map combined. **(b)** Pan UV map filled with textures.

Figure 5.9: Pan UV maps

In addition to the general textures the objects also needed to have bumps as for example the pan had a pattern engraved on its bottom and on the handle. To create the bump maps Adobe Photoshop CC 2019 was used, as it has an option of transforming a picture into a bump map. In the case of thee pan, a picture was taken of the bottom of the pan and the handle, both having engravings. A picture of the engravings on the handle can already be seen in figure 5.4. A screenshot of the picture used to create bump map for the bottom of the pan can be seen in figure 5.10. The process of adding the bumps is the same as adding the normal textures, position them over the UV map. Figure 5.11 shows the entire UV map complete with bumps. Notice the black background of the image. This was done in order to remove artifacts that could appear while applying the bump map in Blender. An example of an artifact can be seen in figure 5.12.

Figure 5.10: Engraving picture used to make the bump map

Figure 5.11: Pan UV map complete with bumps

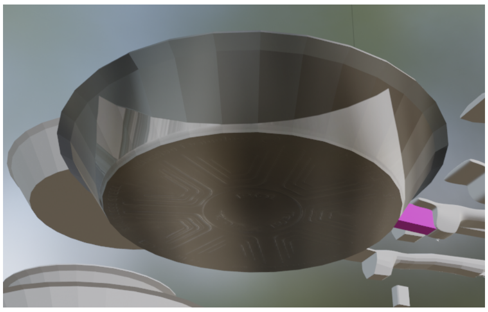


Figure 5.12: Square artifact appearing if background not complete black

Finally after preparing the UV maps the last step is to add them as material in Blender. First step in order to to this is to create a material for the pan and attach textures to that material. Fortunately Blender provides an easy way to do this using nodes. The material configuration for the pan can be seen in figure 5.13. PanTexture4.png is the image file with the material textures, which is simply connected to the base color of the material. For the bump map a special node called "Bump" needs to be introduced. This nodes takes in the imagine file that contains bump map, see figure 5.11. This bump node is connected to the normal property on the main material node of the pan. The different variables for the material are at the default value, the only changes being made to the "Bump" node. It can be seen that the Bump node has a property called "Distance", which was manually set to 0.2, from its default 1. The reason was that 1 provided an effect that was too strong, adding bumps where there would be none. The bumps were also inverted due to them not being treated as engravings otherwise. This is the last step in the texturing process and images of the textured pan can be seen in figure 5.14.

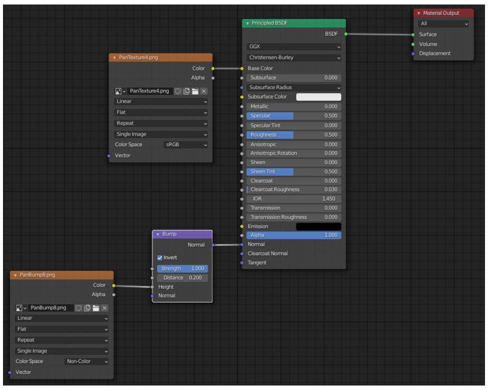
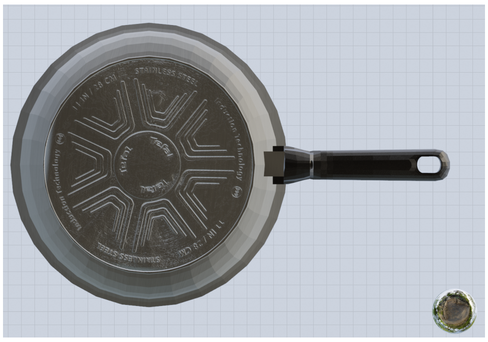
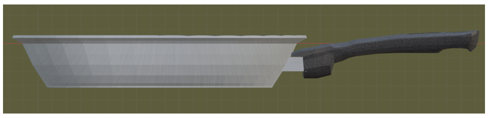


Figure 5.13: Material configuration for the pan in Blender



(a) Pan textured bottom view.

(b) Pan textured top view.



(c) Pan textured side view.

Figure 5.14: Pan textures

The texturing process for the hammer follows the same procedure: make seams on the object, unwrap the UV map, attach the textures to the UV map in image editing software of choice, generate bump images, export to Blender, and as such it will not be detailed here as it would be redundant. To see the UV maps and textures used for the hammer look in Appendix A.

5.3 The software

The software was made using the Unity[4] engine, with the scripts being written in C#. Two scenes have been created, one for each object. Since the two are identical, with the exception of the 3D models being used, only one of them will be described in detail.

5.3.1 The scene

The scene itself is really simple. It consists of the default Unity skydome and a textured plane, that is made to look like a grass field. There is an *OVRCameraRig* object, which replaces the

[4]https://unity.com/

default camera. This is responsible for facilitating the VR based camera, that follows the position and orientation of the Oculus Quest. As a child of this object, there are two *OVRControllerPrefab* objects representing the left and right Oculus controllers respectively. This is necessary in order to visualize the controllers along with the objects, in order not to have a discrepancy between what the users feel with their hands and what they can see. The 3D models of the props are attached to these controller objects as children, in order to facilitate the tracking. The controller and the prop objects have been aligned in a way, that the controller is in the same location relative to the prop as in real life. Also a child of the camera is a *Canvas* object that is used to display the timers. Furthermore there is a manager object, which is an empty game object, with a manger script attached to it. This controls all the functionality in the software.

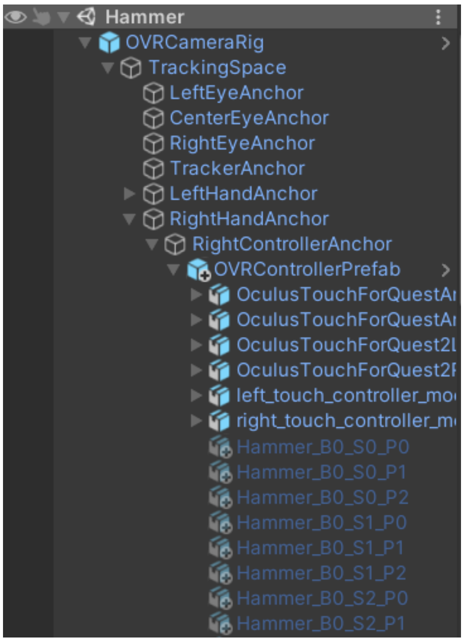


Figure 5.15: The hierarchy from the Unity editor

5.3.2 The objects

Since the objects have certain physical properties that can be altered, different game objects had to be used to represent each version. Since there are three properties that can be modified, each having three levels, there are a total of 27 versions of both the hammer and the frying pan. Each version is stored as a separate game object. The different version of the objects were created partly in Blender and partly in Unity.

(a) Subdivision level 0 **(b)** Subdivision level 1 **(c)** Subdivision level 3

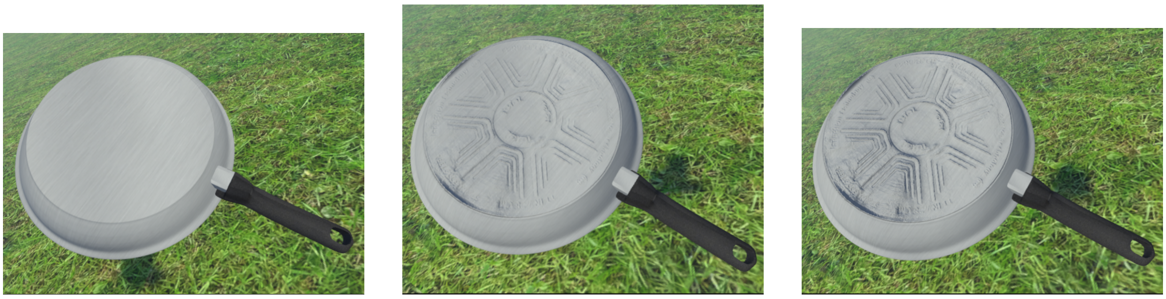
Figure 5.16: The frying pan with three different polygon counts

The polygon count was adjusted in Blender using the *Subdivision* function. The three versions were created using *Subdivision* levels 0, 1 and 3 respectively (see figure 5.16). Bumpiness and size were easier to modify after the objects have been imported to Unity. The 27 objects were created, with the polygon count adjusted on all of them, then the size was adjusted as needed by simply modifying the *Scale* property of the objects in the Unity editor (see figure 5.17). The three levels of size used in the software were real size, in which case the virtual object matches the size of the physical prop, then this size was reduced to 80%, then the resulting size again reduced to 80%.

(a) Small size **(b)** Medium Size **(c)** Real size

Figure 5.17: The frying pan with three different sizes

For the bumpiness, three different material files were created in Unity, each with a different level of the *Bumpiness* property of the normal map of the given texture. The three levels used are 0, 0.15 and 0.25 respectively (see figure 5.18).



(a) Bumpiness level 0 **(b)** Bumpiness level 0.15 **(c)** Bumpiness level 0.25

Figure 5.18: The frying pan with three different bumpiness levels

The current version of the object that is needed to be displayed, is identified by an index, which is a three digit integer, where each digit represents the level of one of the properties. In order to associate each object with an index, a *dictionary* was created, with a *string* being the key, and a *GameObject* being the value in each entry. When the software is launched, the *dictionary* is created, and populated, with the 27 objects. As mentioned in section 4.2, initially the user gets to examine the version of the object that is closest to the physical prop. For this reason, the index is set to be **'222'** by default.

5.3.3 The controller script

Listing 5.1 shows a piece of code that performs the initial setup for each case. A *boolean* called *isFirst* is used to ensure that this piece of code runs only once per case. First a *foreach* loop iterates through the entire *dictionary* and sets all objects to be inactive, thus not visible to the user. This is done so that the object from the previous case gets deactivated, without having to keep track of its index. Following this, the object that is needed for the current case is activated, making it visible. Finally, the timers are reset, along with the *booleans* controlling the clicking and the execution of this setup code.

```
    if (isFirst)
        {
            foreach (KeyValuePair<string, GameObject> entry in hammers)
            {
                entry.Value.SetActive(false);
            }
            timer = timeLimit;
            clickTimer = clickTimeLimit;
            hammers[hammerIndex].SetActive(true);
            firstClick = true;
            isFirst = false;
        }
```

Listing 5.1: *The code that runs only once in each case, performing the setup for the current case*

After the setup is done, the timer for the case starts immediately. As can be seen in listing 5.2 the *demo boolean* keeps track of whether the user is in the initial case or not, as this is a special case. When the timer reaches zero during the initial case, the start button appears. When the user is done inspecting the object and is ready to move on, they have to look at the start button.

```
if (timer <= 0)
        {
            timerDisplay.gameObject.SetActive(false);
            if (demo)
            {
                buttonStart.gameObject.SetActive(true);
                if (Physics.Raycast(cam.transform.position, cam.transform.forward, out hit) && hit.transform.tag == "start")
                {
                    buttonStart.gameObject.SetActive(false);
                    isFirst = true;
                    started = false;
                    demo = false;
                    hammerIndex = startingInd;
                    StreamWriter writer = new StreamWriter(path, true);
                    writer.WriteLine(hammerIndex);
                    writer.Close();
                    bump = int.Parse(hammerIndex.Substring(0, 1));
                    size = int.Parse(hammerIndex.Substring(1, 1));
                    poly = int.Parse(hammerIndex.Substring(2, 1));
                }
```

```
21              }
22              [...]
```

Listing 5.2: *The code that is executed during the initial case*

The button clicking is implemented as gaze based interaction. As line 7 in listing 5.2 shows, there is a raycasting being continuously performed, with the camera being the origin point, and the direction being the forward pointing unit vector of the camera object. The code here checks whether an object tagged **start** is hit by the ray. If it is, that constitutes the clicking of the start button.

After this, the start button gets deactivated, while *isFirst* is set to **true** again, as we are about to enter a new case. The *boolean 'started'* controls whether the timer should be running or not, so that is set to **false** here. *Demo* is another *boolean* being set to **false**, as we are exiting the initial case. The starting index is the next thing to be set up. This is a public variable, which means that its value can be changed directly from the Unity editor window. This value is controlled by the experiment conductors, based on a randomized list of starting indices. After this, the newly established index is written in a log file. The last three lines are where the index is broken down into individual values for each attribute. This is done so that increasing an attribute could be executed by simply incrementing a variable. To extract these values, each time a *substring* is generated from the index, which is of type *string*. A single character is taken from the index, and then parsed into an *int*. *Bump* stores the value for the level of bumpiness of the object, *size* is responsible for tracking the level of size, while *poly* stores the value for the level of polygon count. Each of these variables have a value of 0, 1 or 2.

In listing 5.2 line 4 checks whether the current case is the initial case. If it is not, the controller moves on to the following code, that can be seen in listing 5.3. Here each *if* statement performs a check on the value of one of the variables that store the levels for the adjustable properties. If any of the levels have the value of 2, the corresponding button will not be displayed, as the maximum value for the given property has been reached. Otherwise the buttons are activated in order to be visible for the user.

```
[...]
else

            {
                if (bump < 2)
                {
                    buttonBump.gameObject.SetActive(true);
                }

                if (size < 2)
                {
                    buttonSize.gameObject.SetActive(true);
                }
```

```

                if (poly < 2)
                {
                    buttonShape.gameObject.SetActive(true);
                }
[...]
```

Listing 5.3: *The code that checks whether the buttons need to be displayed*

After the buttons are displayed, the user has the opportunity to click one of them, thus increasing the level of one of the objects physical properties. Since the code that runs after clicking either one of the tree buttons is identical, only one of them will be presented. Listing 5.4 shows the piece of code that runs after the 'bump' button has been pressed.

```
[...]
if (Physics.Raycast(cam.transform.position,
   cam.transform.forward, out hit) && hit.transform.tag == "bump"
   && firstClick)
                {
                    hit.transform.localScale = scaleBig;
                    clickTimerDisplay.gameObject.SetActive(true);
                    int ct = (int)clickTimer;
                    clickTimerDisplay.text = ct.ToString();
                    clickTimer -= Time.deltaTime;

                    if (clickTimer <= 0)
                    {
                        clickTimerDisplay.gameObject.
                        SetActive(false);
                        buttonBump.gameObject.SetActive(false);
                        buttonSize.gameObject.SetActive(false);
                        buttonShape.gameObject.SetActive(false);
                        firstClick = false;
                        started = false;
                        isFirst = true;
                        bump++;
                        hammerIndex = bump.ToString() +
                           size.ToString() + poly.ToString();
                        StreamWriter writer = new
                           StreamWriter(path, true);
                        writer.WriteLine(hammerIndex);
                        writer.Close();
                    }
                }
[...]
```

Listing 5.4: *The code that runs when a button is being clicked*

As for the start button, mentioned in listing 5.2 the button clicking is performed by a raycast, following the same principle. The only difference being, that in order to click one of the buttons, the user has to look at it for three seconds. This is done to prevent accidentally clicking a button. The raycast is again being continuously performed, and in case of a collision with a *GameObject* the tag is examined. The three buttons are identified by their tags, and the corresponding property is incremented accordingly. In line 4, it can be seen that the button the user is looking at, is being scaled up. This is done in order to highlight the button that is about to be clicked. At this point, a timer also starts running, counting down from three. The timer is displayed to the user, and upon reaching zero, the next *if* statement is entered. Upon a click being performed, the timer is deactivated, as well as the buttons. The *booleans* controlling the clicking, timer and the initial setup are reset to their original values, as the user is about to enter a new case. Line 20 shows the variable storing the value of the level of the corresponding property being incremented. In the next line, the new index, that will be used in the next case is generated. Each variable that stores one of the values for the properties is cast to a *string* and assembled, to form the new index. After this the new index is logged in a text file.

```
[...]
else if (!Physics.Raycast(cam.transform.position,
   cam.transform.forward, out hit))
                {
                    clickTimer = clickTimeLimit;
                    clickTimerDisplay.gameObject.SetActive(false);
                    buttonBump.transform.localScale = scaleSmall;
                    buttonShape.transform.localScale = scaleSmall;
                    buttonSize.transform.localScale = scaleSmall;
                }
```

Listing 5.5: *The code that is executed when the user looks away from the buttons*

Listing 5.5 shows what happens when the user looks away from the buttons, before reaching the three seconds required for a click. In this case, the timer is reset, and hidden from the UI, while the buttons get resized to their original size.

Chapter 6

Evaluation

The experiment took place at Aalborg University Copenhagen campus in an isolated room and was conducted in accordance with the method outlined in chapter 3. The equipment used for the experiment is detailed in chapter 4. This chapter will be split into two parts, one detailing the pilot test, and one detailing the actual experiment. Unfortunately because of the COVID pandemic the sample size of the experiment was four participants. Due to this reason the experiment should also be considered a pilot test.

6.1 Pilot test

Before performing the experiment a pilot test was performed in the same location where the experiment took place. This pilot test highlighted some issues that the software was suffering from, and had to be changed before proceeding to the experiment. Initially the time the participants had to look at the ideal configuration was 30 seconds, but it was apparent from the second trial that this was too much time, as the participant got bored waiting for the timer to end. Because of this it was decided to reduce the time to 5 seconds, which is a drastic change, but the participants could use more time if they wanted to, as the trial would not start unless they started it by clicking a button.

Another change to the software was that because it was not apparent enough when a button, that increases a certain property was being clicked, an indicator was needed. It was decided to have the buttons scale up in size a bit whenever the user looked at it. Finally the test participant had a very valuable comment which was that the instructions should be changed to not tell participants that the ideal configuration is the maximum level of all properties. The reason for this is that if the participants know that the ideal configuration is at maximum level, there could be a problem where they would increase the properties to get to this maximum level regardless if they felt like similarity was reached before.

6.2 Experiment

Because the participants did multiple trials the results cannot be truly statistically independent but according to [11] we can make the assumption that they are. Each trial started from a different initial configuration, so the participants had different configurations of properties to choose from every time they performed the trial. This section will present which configurations were discovered to have matched the ideal configuration, the transitions that the participants took getting to those configurations and the responses to the post experiment questionnaire. The section will be split into two parts one detailing the responses and data from the pan trials, the other one detailing the responses from the hammer trials.

6.2.1 Participants

The participants were all taken from the campus at Aalborg University Copenhagen. Their mean age was $\mu = 26.75$ years with SD (standard deviation) = 2.87. For most of them, the highest finished level of education was a Bachelor's degree. All participants used VR headsets before, and also all of them used physical props in VR.

6.2.2 Pan

This part will detail the data gathered from the pan trials in terms of accepted configurations, transitions and questionnaire responses.

6.2.2.1 Accepted configurations

As mentioned in chapter 3 the task the participant had to do was to improve physical properties until they feel that they are in the ideal configuration, which is Similarity = $\{222\}$. Figure 6.1 shows the probabilities for the accepted configurations. The blue line represents the percentage that each configuration makes up of all accepted configurations (absolute probability). The orange line represents the probability that a configuration was accepted as being matching once it was reached (conditional probability). So for example the case '122' was accepted 7 times out of a total of 16 accepted configurations, which means a probability of 0.43. However it was visited 8 times while being accepted 7 times, which amounts to a probability of 0.87 to be accepted when reached. Case '120' was accepted 1 time out of 16 total configurations amounting to a probability of 0.062. It was visited 7 times and got accepted once, leading to a to a probability of 0.14. Case '122' got accepted 7 times out of total 16 which is a probability of 0.43. It was also accepted 7 out of 8 times it was visited, which is a probability of 0.87. Case '212' was reached once out of 16 total with a probability of 0.062. The only time this case was visited it also got accepted meaning it had a probability of 1 to be accepted when reached. Case '222' was accepted 2 times out of 16 total, which means a probability of 0.14. Of course being the last configuration it was also accepted every time it was visited meaning a probability of 1. The average number of improvements the participants did while performing the trials for the pan is 4. This can be seen in figure 6.2 which shows configurations that were accepted the most amount of times. Configuration '121' was accepted 5 times and configuration '122' was accepted 7 times.

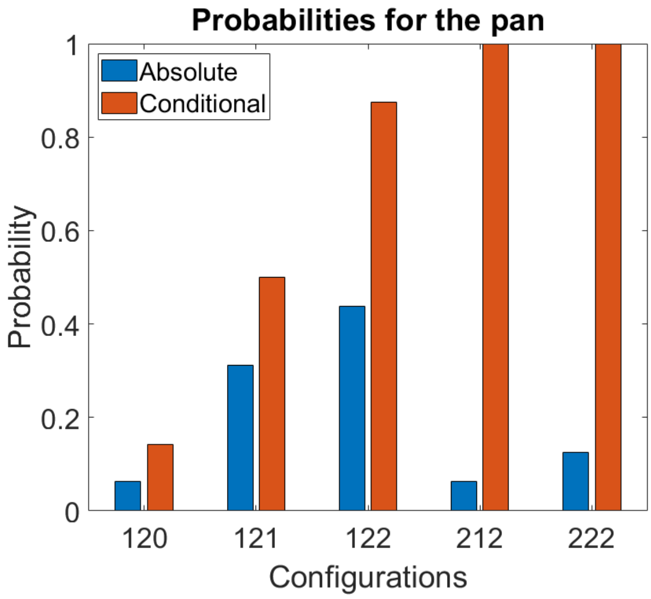


Figure 6.1: Accepted configurations and their probabilities. The blue bar represents the probability of accepting a configuration, meaning when participants reported a match for the configuration over the total number of configurations. The orange bar represents the probability that a configuration will be accepted when reached

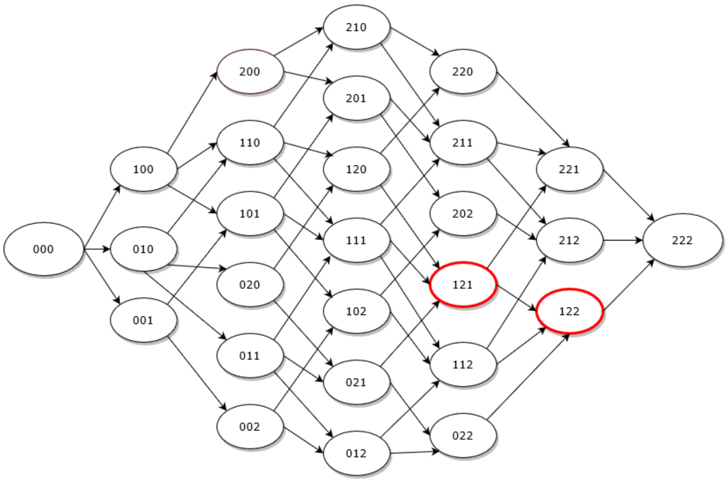


Figure 6.2: Configurations that were accepted the most

6.2.2.2 Transitions

Figure 6.3 shows the transition probability diagram for the trials performed with the pan. The probability transition diagram was constructed to discover what the most chosen path would be.

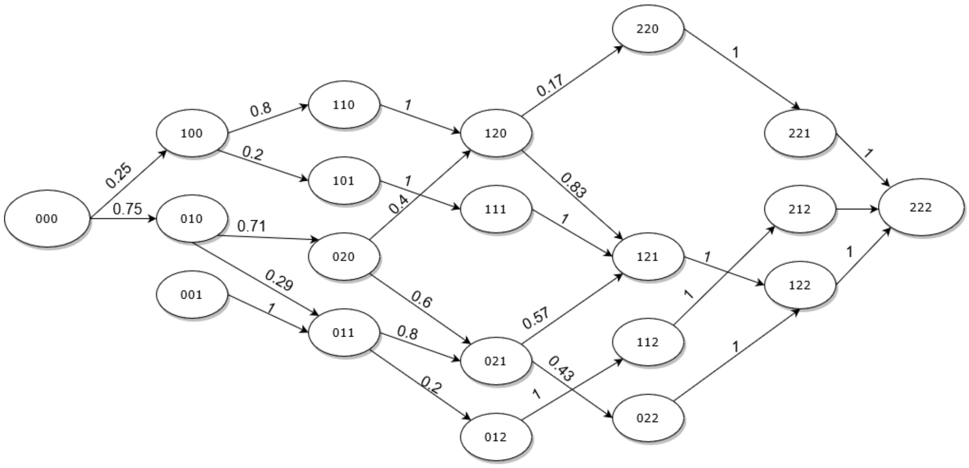


Figure 6.3: Transition probability diagram for the pan

Figure 6.4 shows the most taken path through the diagram regarding the pan trials. As mentioned earlier in the report the configurations are denoted by the vector Similarity = {Surface texture Size Shape}. It can be seen from the figure that the majority of participants decided to upgrade the size property twice leading to configuration '020'. After that the majority decided to improve the shape and bring it to level 1 leading to '021'. Following this, they went on to improve surface texture, leading to configuration '121'. From this node they improved the size to level 2 leading to configuration '122'. Finally from this node there is only one choice and that is improving the surface texture to level 2 leading to configuration '222'.

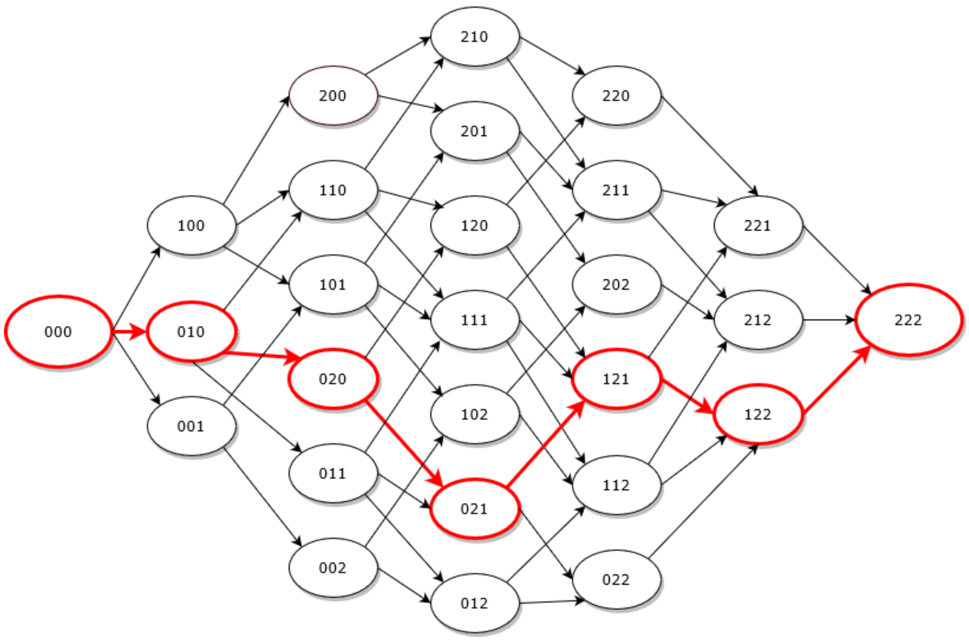


Figure 6.4: The most likely path through the graph, in the case of the pan

6.2.2.3 Questionnaires

The questionnaire for the pan can be seen in Appendix B figure B.1. The participants were asked to rate how much the given property contributed to the feeling of similarity between the physical and virtual objects on a 5 point Likert scale, and to explain why they rated it as they did. Figure 6.5 shows the box plot for each of the 3 properties. It seems that the most important property in case of the pan is the size property with a median of M = 5 and a standard deviation of SD = 0.50. The median and standard deviation for shape and surface texture in the case of the pan is the same. Both medians being M = 3.5 and standard deviation being SD = 0.95. The comments from these questionnaires will be presented in chapter 7.

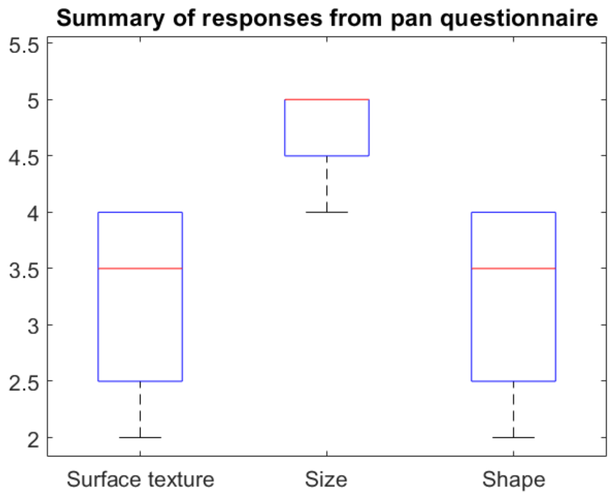


Figure 6.5: Box plot of the questionnaire responses concerning the pan

6.2.3 Hammer

This section will present the data gathered from the trials that included the hammer. The same format will be used as in the previous section related to the pan.

6.2.3.1 Accepted configurations

Figure 6.6 presents the probabilities for the hammer trials. The representation is the same as in section 6.2.2.1 with absolute and conditional probabilities. Case '012' was accepted once out of 16 total accepted configurations which means a probability of 0.62. It was also accepted only once, while being reached 7 times, which is a probability of 0.14 to be accepted when reached. The probabilities for case '021' are the same as '012'. Case '121' was accepted 9 times out of 16 total accepted configurations, which leads to a probability of 0.56 to be accepted. It was visited 13 times and accepted 9 times which means a probability of 0.69 when reached. Case '122' was accepted 4 times which leads to a probability of 0.25 to be accepted. It was also visited 4 times which amounts to a probability of 1 to be accepted when reached. Case '221' was accepted once, which means a probability of 0.062 to be accepted. It was visited once, which gives it a probability of 1 to be accepted when reached. The average number of improvements for the hammer trials is 3.4, which again corresponds to the two most accepted configurations in figure 6.2

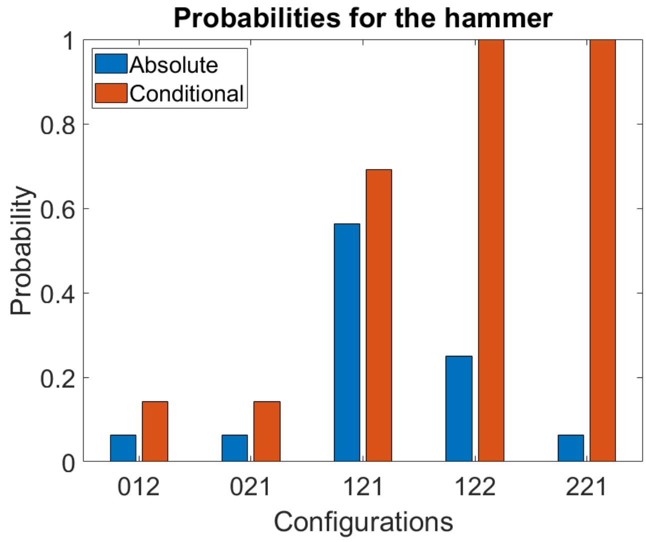


Figure 6.6: Accepted configurations and their probabilities. The blue bar represents the probability of accepting a configuration, meaning when participants reported a match for the configuration over the total number of configurations. The orange bar represents the probability that a configuration will be accepted when reached

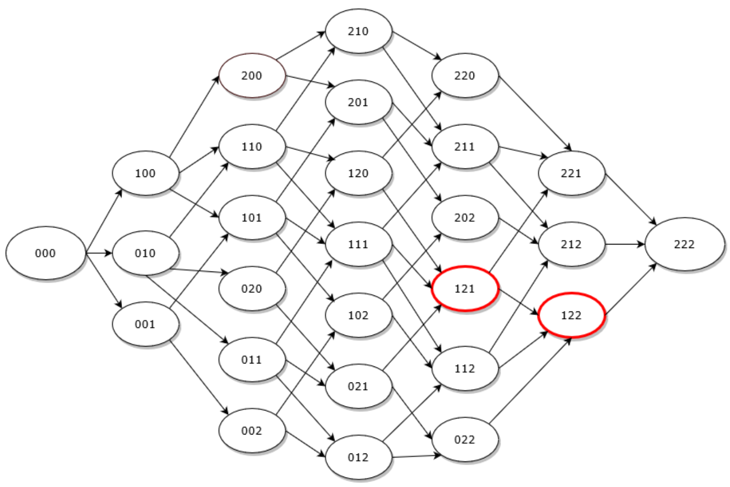


Figure 6.7: Configurations that were accepted the most

6.2.3.2 Transitions

Figure 6.8 shows the transition probability diagram for the hammer.

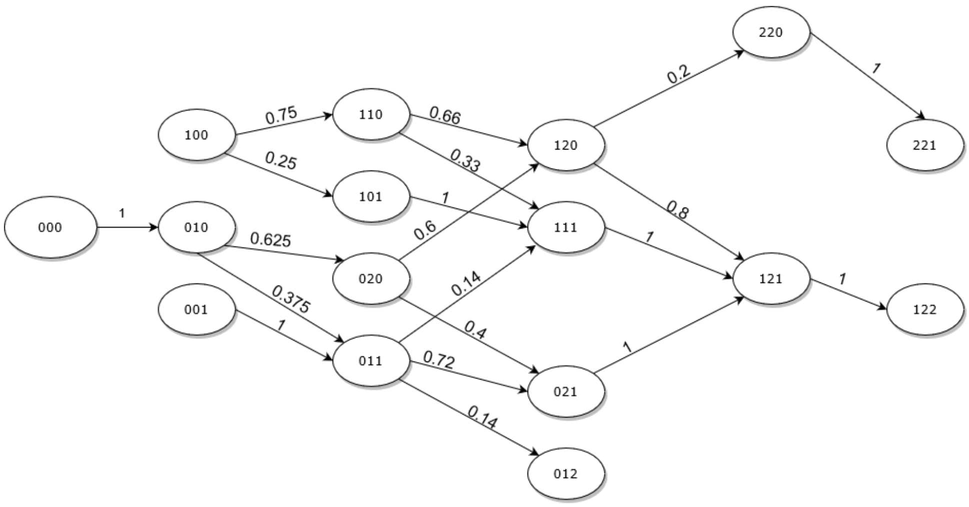


Figure 6.8: Transition probability diagram for the hammer

Figure 6.9 details the most taken path through the diagram in the case of the hammer. Again, as it was the case for the pan most participants decided to improve the size twice leading to configuration '020'. Following that the majority decided to upgrade the surface texture once, leading to configuration '120'. After the texture they improved the shape leading to '121'. Finally majority decided to improve the shape a second time leading to configuration '122'. In the case of the hammer there was no participant that reached the '222' configuration.

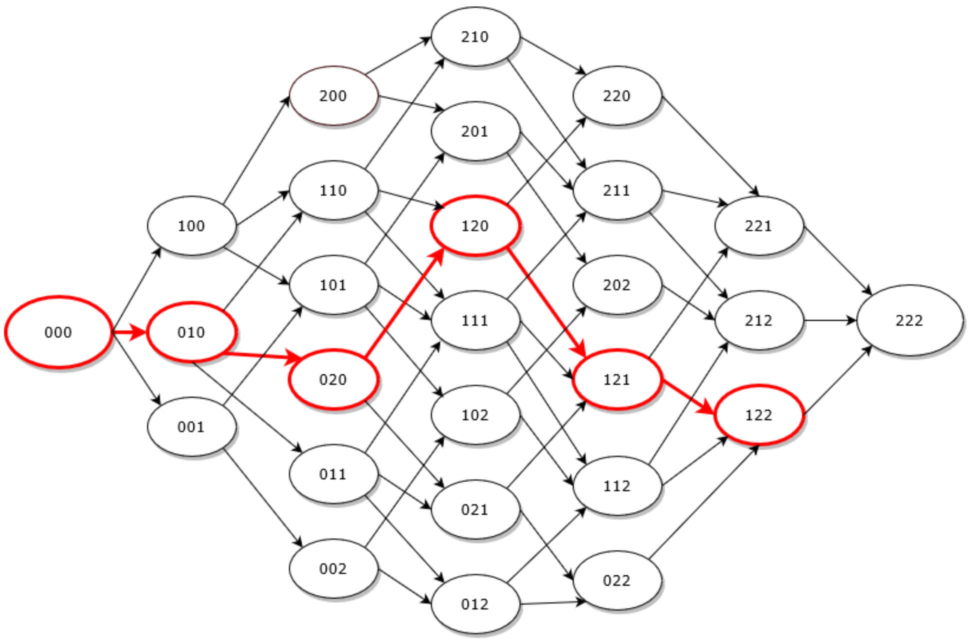


Figure 6.9: The most likely path through the graph, in the case of the hammer

6.2.3.3 Questionnaire

The questionnaire for the hammer can be seen in Appendix B figure B.2. Figure 6.10 shows a box plot each of the 3 properties for the hammer. It can be seen that it is mostly the same with size having the highest median of M = 5 and SD = 0.5. Surface texture had a median M = 2 and SD = 1 while shape had a median M = 3 and a SD = 0.81. Again, comments about subjective responses from the survey will be talked about in discussion.

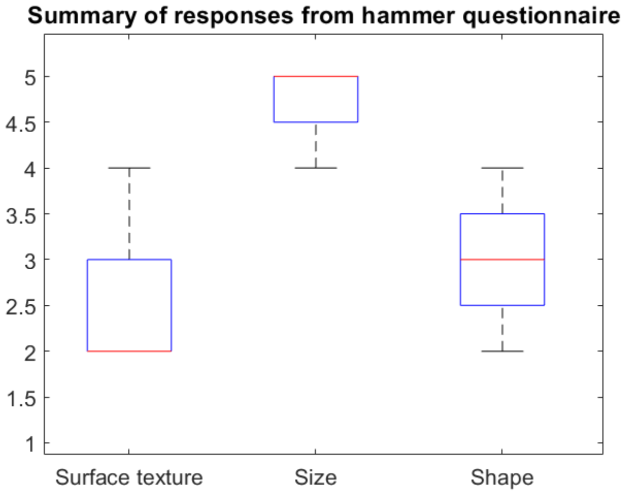


Figure 6.10: Box plot of the questionnaire responses concerning the pan

Chapter 7

Discussion

In this section the previously presented results will be discussed in detail. However, it should be noted that due to the small sample size, and this being treated as a pilot test, no actual conclusions can be drawn from the data. This section serves merely as a representation of how the data would be analyzed in case the study was run under normal circumstances with a larger sample size.

7.1 Size is the most important property

Size seems to be the most important property to improve. As it can be seen in figures 6.4 and 6.9 the majority of test participants decided to improve the size property twice as soon as the trial started. It should be noted that there is a difference in preferred improvements after reaching size level 2. In the case of the pan the majority chose to improve the shape next, while in the case of the hammer the majority chose to improve the surface texture. However, the reason for this cannot be determined from the available data. Figure 6.3 shows the transition probability diagram for the pan, and it can be seen that no matter the initial case, the majority of participants decided to improve the size at least once. The same can be said in the case of the hammer trials, as seen in figure 6.8. It can also be seen in figures 6.1 and 6.6 that most of the accepted configurations had size at level 2. In fact there is a probability of 0.93 that the accepted configurations had size at level 2 both in the case of the hammer and the pan. Figures 6.2 and 6.7 show configurations that were accepted the most amount of times, both of them having size at level 2. Figures 6.5 and 6.10 both show that the highest medians belong to the size property. The standard deviation is also lowest in the case of the size property.

Regarding the subjective responses to the questionnaires, the majority of people explained that the size is important because when the virtual object was in lower levels there was a discrepancy between the weight of the physical object they were holding and the size of the virtual object they were seeing. This is an important comment as it shows that the perception of the size of an object in virtual reality is tied to its weight in physical space, and discrepancies are apparent when the size is different. This was true in both the case of the pan and the hammer. The responses from the questionnaire can be seen in Appendix B figures B.3 and B.4. The importance of size in mismatching physical props and virtual items is also mentioned in the paper written by Simenone et al. [9].

7.2 Surface texture and shape can remain at level 1

As it can be seen in figures 6.1 and 6.6, in the case of the pan, there was only one time when a configuration was accepted and it did not have the three properties at level 1 at least. This means a probability of 0.0625 that a configuration will be accepted that has one attribute at level 0. In the case of the pan the configuration that was accepted was '120'. In the case of the hammer there were only two times when a configuration was accepted, that did not have all properties at level 1 at least, which means a probability of 0.125. It can be seen from figures 6.2 and 6.7 that configuration '121' is one of the most accepted in both the case of the pan and the hammer. Even though the sample size was small, the median and standard deviations calculated from the questionnaire responses also hint at the importance of the other two properties being more or less the same. This can be seen in figures 6.5 and 6.10, where the median and standard deviations for surface texture and shape properties are the same in case of the pan and slightly different in the case of the hammer (responses were recorded on a 5 point scale). For the pan the median and standard deviation for both attributes is M = 3.5 and SD = 0.95. For the hammer, surface texture has a mean of M = 2 and SD = 1, while shape has a mean of M = 3 and SD = 0.81.

The responses from the questionnaire as to why they rated the surface texture and shape as they did in the case of the pan can be seen in Appendix B figures B.5 and B.7. It seems as they considered the surface texture less important because the property controlled small details on the pan, and it only mattered if that was very off, smaller details not being too important. This reinforces the idea that the surface texture property can be less than ideal and it will still give the same sense of similarity as a more detailed level of this property.

When rating the shape property and detailing why they rated it so, participants mentioned that the "ridged" geometry looked fake. This ridged geometry most likely refers to the low polygon count of the object, which is visually apparent. One participant mentioned that they saw that the graphics were a bit low resolution, possibly meaning that they could see the polygons clearly on the object. There was also one participant that mentioned that the biggest change was between level 0 and 1 regarding the shape of the pan, and that this configuration would suffice. It seems that when it comes to the pan, a smaller polygon count can produce the same sense of similarity as a higher polygon count.

The responses from the questionnaire related to surface texture and shape while handling the hammer can be seen in Appendix B, figures B.7 and B.8. From the responses concerning surface texture the reasons for why participants did not consider this property as important seem to be the same as in the case of the pan, the details were small and a less pronounced bump map was enough. Regarding the shape it seems that level 1 was enough because it rounded off the handle of the hammer, which was enough to be considered ideal. The difference between shape level 1 and 2 is not that apparent as the polygon count is large enough on level 1 that the hammer's handle looks round.

As mentioned earlier in this chapter, the participants chose to improve properties in a different order for each item. This suggests, that the physical properties of the prop might influence what is considered acceptable and what is not. Therefore, in case the study is repeated with the proper sample size, the use of more items is recommended. Items with different global shapes, sizes and weights should be used, in order to discover whether these differences influence the users' decisions in any way.

Chapter 8

Conclusion

As the final problem statement mentioned, this project investigated:

Which of the following physical properties: surface texture, size, shape is most important when trying to match physical props with their virtual counterpart and what level of mismatch is tolerable?

In order to try to answer this question, we have developed a VR application, where physical props could be compared with virtual objects, that had different settings for some of their physical properties (surface texture, size, shape). A user study was designed to test how much difference in the aforementioned properties is tolerated between the physical and the virtual objects. However, due to the currently ongoing pandemic situation the project was severely limited in its scope. Ideas and features had to be scrapped, and sub-optimal solutions had to be used in order to implement our ideas. In the end, the implementation was finished, but running the study was not possible the way it was originally envisioned. Due to not having access to many people, the number of participants ended up being extremely low, thus the experiment had to be downgraded to a pilot test. That said, some trends were still visible from this low sample size, which we tried to analyze as best as possible. It is important to note however, that no real conclusions can be drawn from the data collected. With that in mind, a preference for the size attribute could be observed, with the other properties requiring only minimal adjustments from their initial state in order for the users to believe that physical prop they are holding, and the virtual object they are seeing is the same. Based on these findings, the experiment could be perfected, and repeated under normal circumstances, in order to get real answers to the question at hand.

Chapter 9

Future works

As mentioned before, due to the small sample size this experiment is treated as a pilot test by the project group. Therefore, this section will be used to describe how the experiment should proceed if repeated under normal circumstances, meaning having full access to the University's facilities, and an adequate amount of test participants. Furthermore, ideas that were originally envisioned, but were scrapped on account of the pandemic, will also be presented.

9.1 Original ideas

At the beginning of the project, we had imagined a much larger scale implementation and experiment. The idea was, that in order to help design props for use in virtual reality, we would investigate how similar a prop has to be to the object it represents, in order for the user to believe that they are the same.

A proper way to test this would be to design a prop ourselves, make a 3D model of it, then manufacture the prop in several instances. Each instance of the prop would have different material properties, as well as geometric properties. For example, the props would be of different size, shape and maybe even material. This way, a threshold for believability could be found, that could then be used to simplify prop design and production in the future. It is also possible, that new prop design principles could be laid down as a result of that experiment.

Another scrapped idea was to make the users perform a small task with the objects. This could be relevant, because physical props in VR are meant for interactivity, therefore their similarity to their virtual counterparts should also be examined in an interactive scenario. We had planned to ask the users to perform a simple task, that would be organic to the object they are handling, and make them judge similarity that way.

9.2 Running the study under normal circumstances

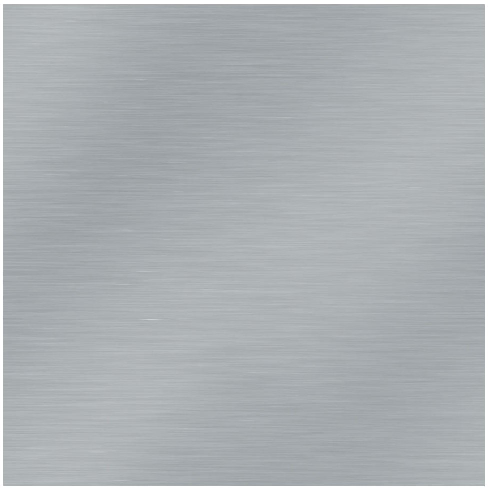
If it were not for the COVID-19 pandemic, we would have been able to run a much larger scale study. That said, in order to get ample amount of data, the study should be repeated once the conditions allow for that. The sample size should be much greater, which would not be a problem to achieve under normal circumstances.

Another improvement to the study could be the use of better tracking equipment. Either optical tracking, as mentioned in [9] or dedicated trackers, such as the HTC Vive trackers, that we were originally planning to use. This way, the tracking would not interfere with the users' perception of the props, and they would not be limited in terms of what they can do with the props when examining them. Using the dedicated trackers would also ensure a more solid attachment of the trackers to the objects.

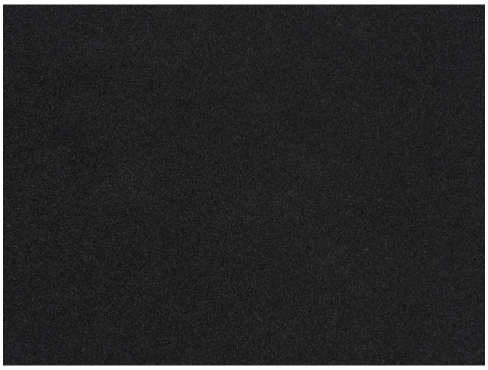
Appendices

Appendix A

Figure A.1: Hammer seams

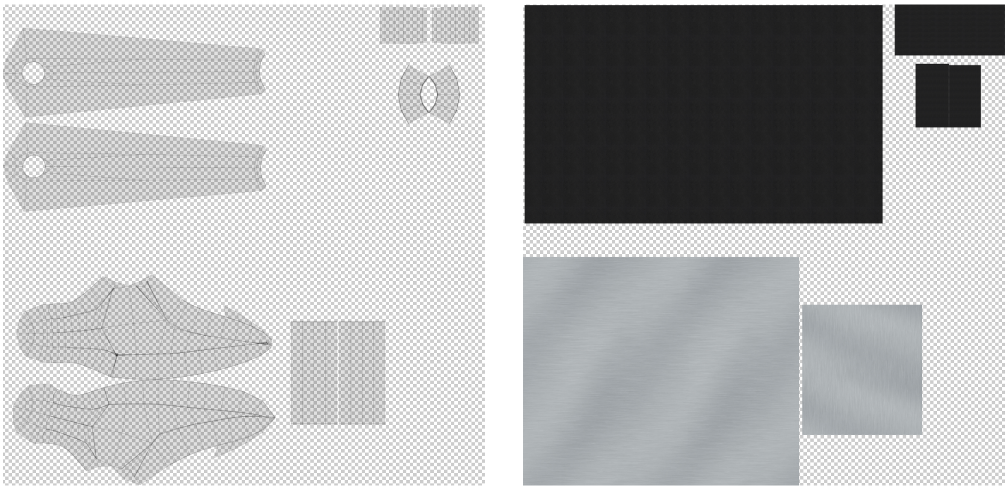


(a) Steel texture used for head.



(b) Plastic texture used for handle.

Figure A.2: Hammer textures



(a) Hammer UV map not filled. **(b)** Hammer UV map filled.

Figure A.3: Hammer UV maps filled and not filled

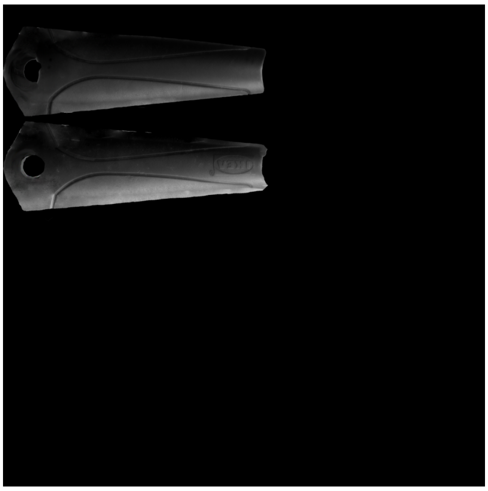


Figure A.4: Hammer bump

Figure A.5: Hammer side view textured

Appendix B

2 - Questions about the frying pan scenario

Description (optional)

How much did you feel like 'Surface texture' contributed to the feeling of the virtual frying pan being similar to the real frying pan? *

	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution

Why? *

Short-answer text

How much did you feel like 'Size' contributed to the feeling of the virtual frying pan being similar to the real frying pan? *

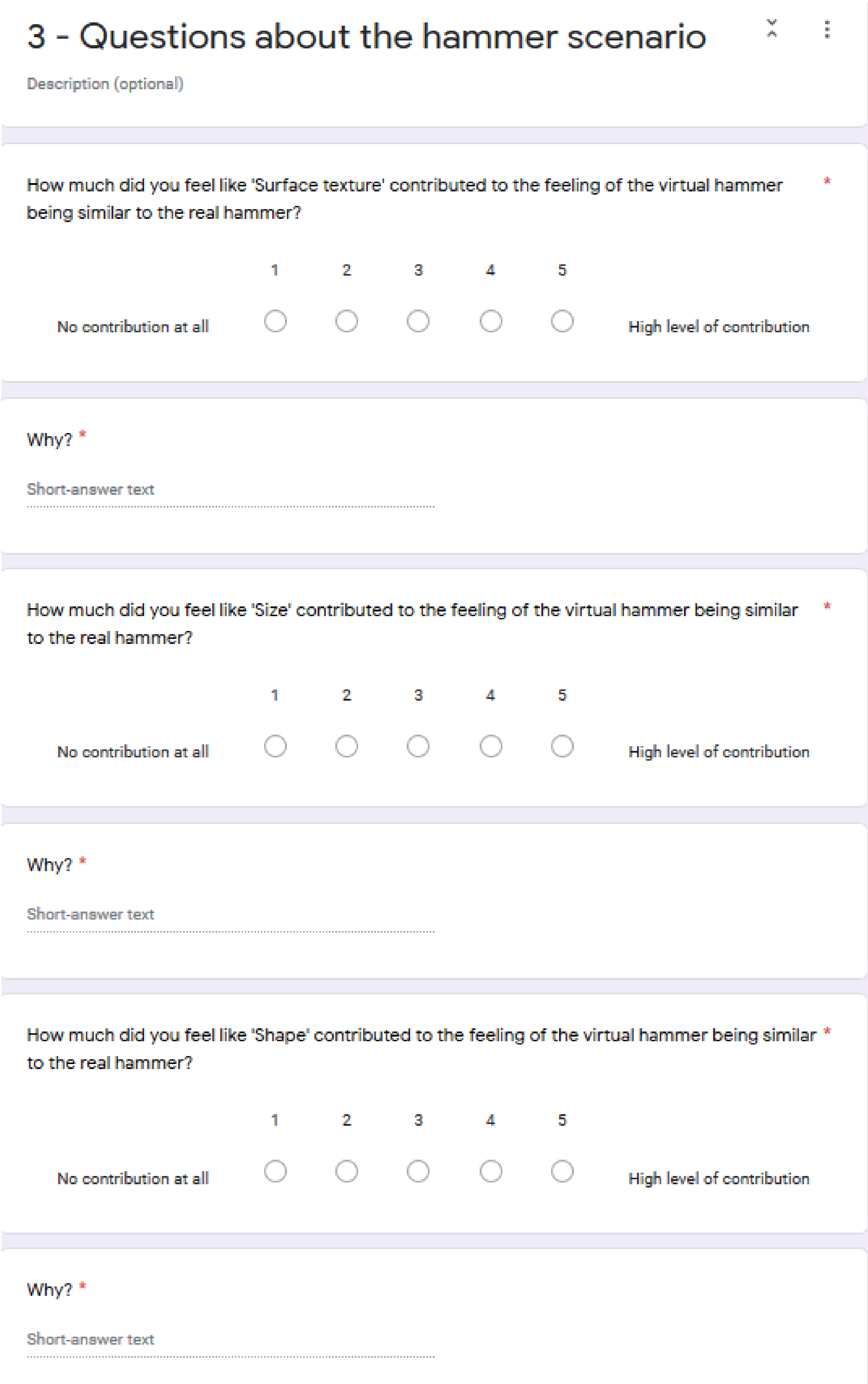
	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution

Why? *

Short-answer text

How much did you feel like 'Shape' contributed to the feeling of the virtual frying pan being similar to the real frying pan? *

	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution



3 - Questions about the hammer scenario

Description (optional)

How much did you feel like 'Surface texture' contributed to the feeling of the virtual hammer being similar to the real hammer? *

	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution

Why? *

Short-answer text

How much did you feel like 'Size' contributed to the feeling of the virtual hammer being similar to the real hammer? *

	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution

Why? *

Short-answer text

How much did you feel like 'Shape' contributed to the feeling of the virtual hammer being similar to the real hammer? *

	1	2	3	4	5	
No contribution at all	○	○	○	○	○	High level of contribution

Why? *

Short-answer text

Figure B.2: Questionnaire part about the hammer

Why?

feeling of physical object size and weight did not match the virtual one if it was too small

1 response

It was very apparent if the size didnt match the physical size and percieved weight of the object.

1 response

bbecause the size did not coincide with the weight

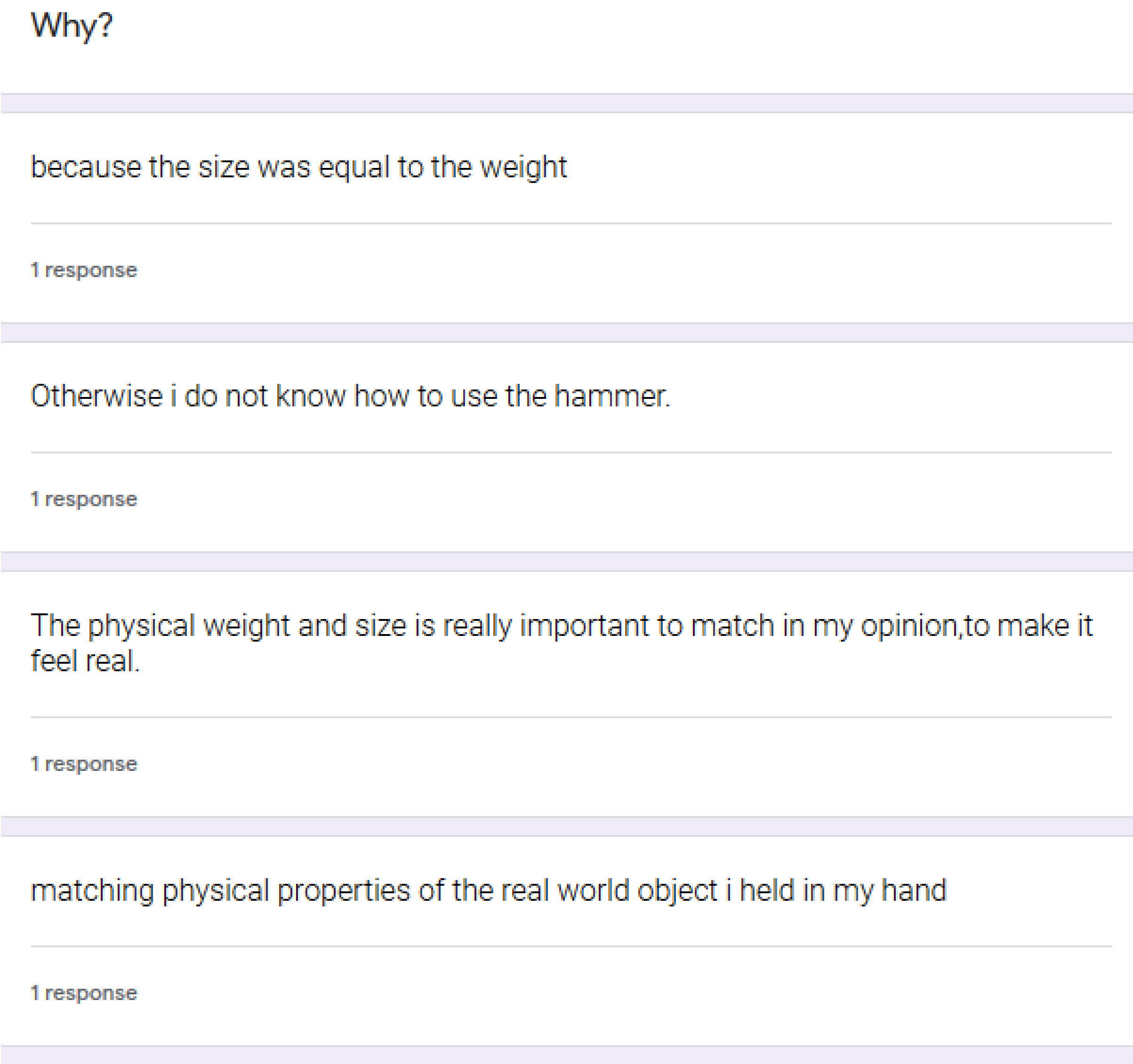
1 response

Size is super important for me. I loose all connection between VR and the real world.

1 response

‹ ›

Figure B.3: Responses to the questionnaire related to the pan as to why they rated the importance of the size property as they did


Why?

because the size was equal to the weight

1 response

Otherwise i do not know how to use the hammer.

1 response

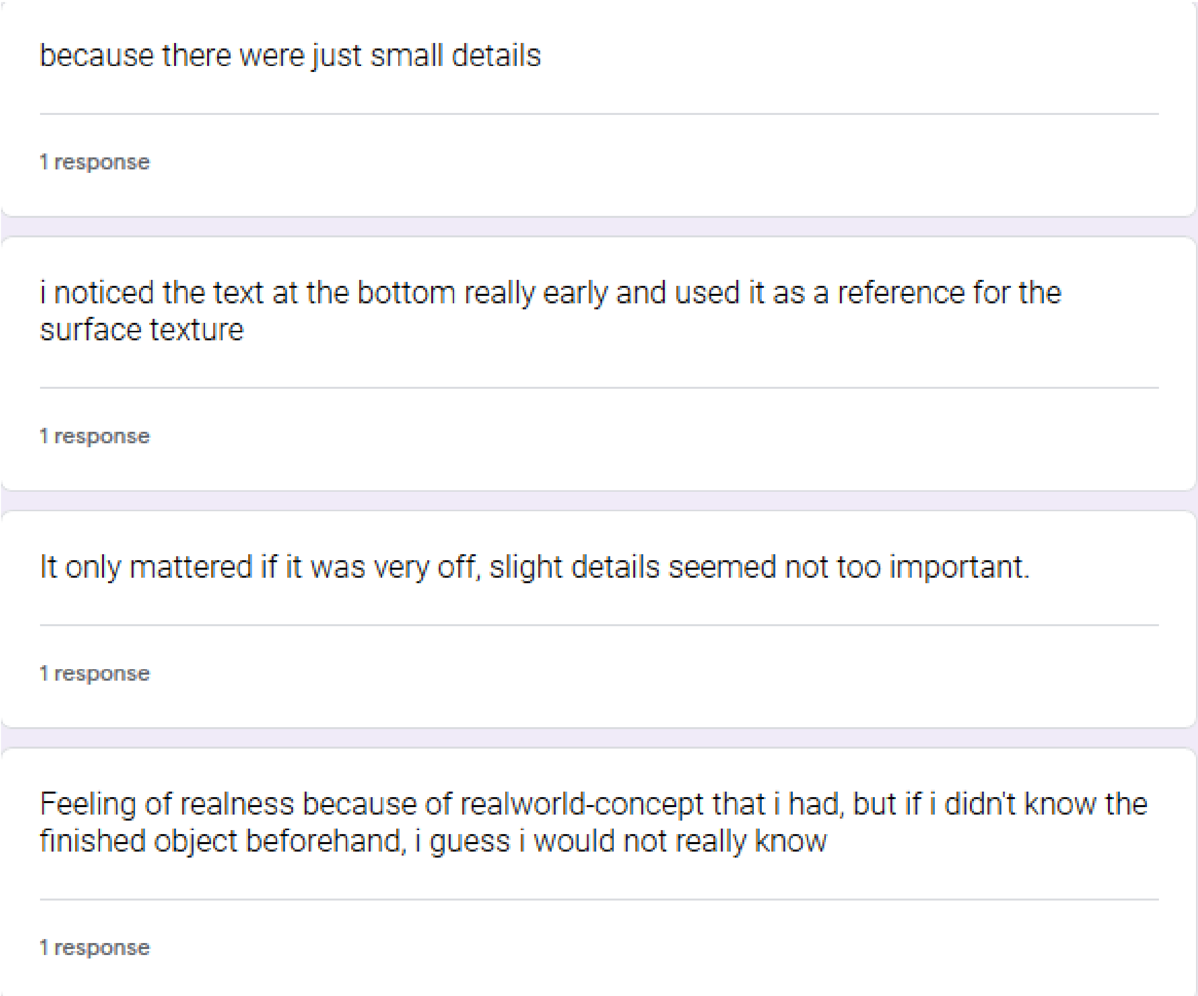
The physical weight and size is really important to match in my opinion,to make it feel real.

1 response

matching physical properties of the real world object i held in my hand

1 response

Figure B.4: Responses to the questionnaire related to the hammer as to why they rated the importance of the size property as they did



because there were just small details

1 response

i noticed the text at the bottom really early and used it as a reference for the surface texture

1 response

It only mattered if it was very off, slight details seemed not too important.

1 response

Feeling of realness because of realworld-concept that i had, but if i didn't know the finished object beforehand, i guess i would not really know

1 response

Figure B.5: Responses to the questionnaire related to the pan as to why they rated the importance of the surface texture property as they did

When the geometry is "riged" it looks really fake

1 response

because the shape made more real the object

1 response

depending on what i know beforehand, the middle configurations would suffice, and there was greatest difference between lowest and the next level, so in theory i could have a match before reaching the highest level, but again depending on what i knew the final object to be

1 response

I could see that the graphics were a bit low resolution, however it didnt make think too much if it was very similar to the real world object

1 response

Figure B.6: Responses to the questionnaire related to the pan as to why they rated the importance of the shape property as they did

Why?

the texture made up more of the hammer object than the pan *which was primarily seen in the bottom, and now here at the shaft

1 response

I did not notice any texture on the tip of the hammer

1 response

not that much because there were small details

1 response

Mostly same reason why previously, and also there were less physical details on the hammer I think.

1 response

Figure B.7: Responses to the questionnaire related to the hammer as to why they rated the importance of the shape property as they did

Why?

because the smallest the hammer was the less weight had

1 response

To some extent yes, mostly at the end of the hammer if the graphics were not curved, then it felt quite off.

1 response

neglicable, but i noticed the pointy end in lower level settings, wanting to change that. but that was the only feature on the hammer that mattered to me

1 response

Very little unless at the tip

1 response

Figure B.8: Responses to the questionnaire related to the hammer as to why they rated the importance of the shape property as they did

Bibliography

[1] F. Daiber, D. Degraen, A. Zenner, F. Steinicke, O. J. Ariza Núñez, and A. L. Simeone, "Everyday proxy objects for virtual reality," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–8, ISBN: 9781450368193. DOI: 10.1145/3334480.3375165. [Online]. Available: https://doi.org/10.1145/3334480.3375165.

[2] A. Zenner and A. Krüger, "Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 4, pp. 1285–1294, 2017. DOI: 10.1109/TVCG.2017.2656978.

[3] B. E. Insko, M. Meehan, M. Whitton, and F. Brooks, "Passive haptics significantly enhances virtual environments," Ph.D. dissertation, Citeseer, 2001.

[4] R. D. Joyce and S. Robinson, "Passive haptics to enhance virtual reality simulations," in *AIAA Modeling and Simulation Technologies Conference*, 2017, p. 1313.

[5] P. L. Strandholt, O. A. Dogaru, N. C. Nilsson, R. Nordahl, and S. Serafin, "Knock on wood: Combining redirected touching and physical props for tool-based interaction in virtual reality," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13, ISBN: 9781450367080. DOI: 10.1145/3313831.3376303. [Online]. Available: https://doi.org/10.1145/3313831.3376303.

[6] N. C. Nilsson, A. Zenner, and A. L. Simeone, "Haptic proxies for virtual reality: Success criteria and taxonomy," in *Workshop on Everyday Proxy Objects for Virtual Reality at CHI'20*, 2020.

[7] M. Azmandian, M. Hancock, H. Benko, E. Ofek, and A. D. Wilson, "Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences," in *Proceedings of the 2016 chi conference on human factors in computing systems*, 2016, pp. 1968–1979.

[8] A. Zenner and A. Krüger, "Drag:on: A virtual reality controller providing haptic feedback based on drag and weight shift," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–12, ISBN: 9781450359702. DOI: 10.1145/3290605.3300441. [Online]. Available: https://doi.org/10.1145/3290605.3300441.

[9] A. L. Simeone, E. Velloso, and H. Gellersen, "Substitutional reality: Using the physical environment to design virtual reality experiences," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15, Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 3307–3316, ISBN: 9781450331456. DOI: 10.1145/2702123.2702389. [Online]. Available: https://doi.org/10.1145/2702123.2702389.

[10] J. Bergström, A. Mottelson, and J. Knibbe, "Resized grasping in vr: Estimating thresholds for object discrimination," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '19, New Orleans, LA, USA: Association for Computing Machinery, 2019, pp. 1175–1183, ISBN: 9781450368162. DOI: 10.1145/3332165.3347939. [Online]. Available: https://doi.org/10.1145/3332165.3347939.

[11] R. Skarbez, S. Neyret, F. P. Brooks, M. Slater, and M. C. Whitton, "A psychophysical experiment regarding components of the plausibility illusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 4, pp. 1369–1378, 2017. DOI: 10.1109/TVCG.2017.2657158.

[12] M. Slater, B. Spanlang, and D. Corominas, "Simulating virtual environments within virtual environments as the basis for a psychophysics of presence," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 4, pp. 1–9, 2010.

[13] B. Hannaford and A. M. Okamura, "Haptics," in *Springer Handbook of Robotics*, Springer, 2016.

[14] S. Lederman and R. Klatzky, "Designing haptic and multimodal interfaces: A cognitive scientist's perspective," in *Proc. of the Workshop on Advances in Interactive Multimodal Telepresence Systems*, 2001, pp. 71–80.

[15] H. Culbertson, S. B. Schorr, and A. M. Okamura, "Haptics: The present and future of artificial touch sensation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 385–409, 2018.

[16] K. O. Johnson, T. Yoshioka, and F. Vega–Bermudez, "Tactile functions of mechanoreceptive afferents innervating the hand," *Journal of Clinical Neurophysiology*, vol. 17, no. 6, pp. 539–558, 2000.

[17] R. W. Lindeman, J. L. Sibert, and J. K. Hahn, "Hand-held windows: Towards effective 2d interaction in immersive virtual environments," in *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, IEEE, 1999, pp. 205–212.

[18] B. E. Insko, M. Meehan, M. Whitton, and F. Brooks, "Passive haptics significantly enhances virtual environments," Ph.D. dissertation, Citeseer, 2001.

[19] S. Razzaque, Z. Kohn, and M. C. Whitton, *Redirected walking*. Citeseer, 2005.

[20] L. Kohli, E. Burns, D. Miller, and H. Fuchs, "Combining passive haptics with redirected walking," in *Proceedings of the 2005 international conference on Augmented tele-existence*, 2005, pp. 253–254.

[21] L. Kohli, "Exploiting perceptual illusions to enhance passive haptics," in *IEEE VR Workshop on Perceptual Illusions in Virtual Environments*, Citeseer, 2009, pp. 22–24.

[22] M. Achibet, A. Girard, A. Talvas, M. Marchal, and A. Lécuyer, "Elastic-arm: Human-scale passive haptic feedback for augmenting interaction and perception in virtual environments," in *2015 IEEE Virtual Reality (VR)*, IEEE, 2015, pp. 63–68.

[23] J. Bergström, A. Mottelson, and J. Knibbe, "Resized grasping in vr: Estimating thresholds for object discrimination," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, 2019, pp. 1175–1183.

[24] R. Skarbez, "A preliminary investigation of place illusion and plausibility illusion," *IEEE Virtual Reality (VR) Doctoral Consortium*, 2015.

[25] A. Lécuyer, S. Coquillart, A. Kheddar, P. Richard, and P. Coiffet, "Pseudo-haptic feedback: Can isometric input devices simulate force feedback?" In *Proceedings IEEE Virtual Reality 2000 (Cat. No. 00CB37048)*, IEEE, 2000, pp. 83–90.