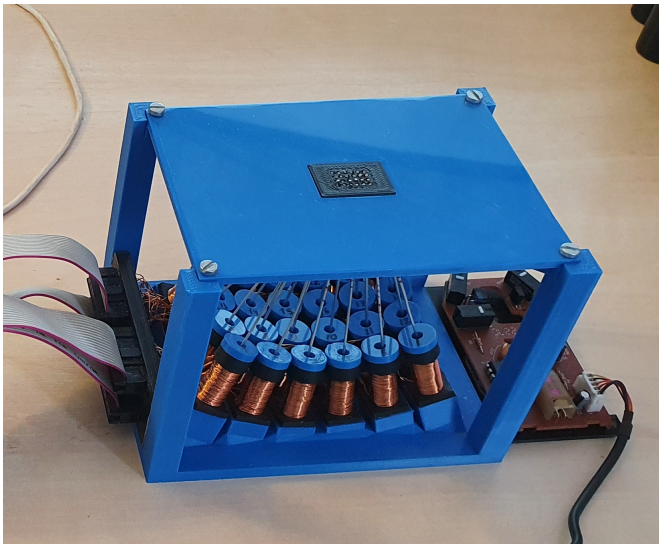
FINGERTIP HAPTIC TEXTURE DEVICE



FINGERTIP HAPTIC TEXTURE DEVICE

Masters Thesis

by

Jelger LEMMERS

in partial fulfillment of the requirements for the degree of
Master of Science
in **Embedded Systems**
at Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science,
to be defended publicly on **January 29, 2021 at 10:00 AM**

Student number: 4721462
Project duration: Nov. 25, 2019 - Jan. 29, 2021
Thesis committee: Dr. Venkatesha Prasad
Dr. Raj Thilak Rajan
Dr. Vijay Rao
Ir. Kees Kroep
Dr. Vineet Gokhale

ABSTRACT

Tactile internet promises to enable humans to manipulate remote physical environments. The human interacts with the remote environment by manipulating a haptic device. Recently, research effort into tactile internet and haptic devices have seen a significant increase. However replicating one aspect of executing an action, feeling textures, is still in its infancy. Current devices that can portray texture are severely limited, which also interferes with the ability to research the topic. In this thesis, we propose a novel device that can realistically portray textures to the fingertip, and respond to displacement in real-time. We call this a Haptic Texture Device (HTD). The design features custom actuators, with low latency < 0.6 ms and at least 50 mN per mm^2 of pushing force. The actuators are placed in a grid with a 2.0 mm pitch at the fingertip. The Haptic Texture Device (HTD) can be moved over a 2D surface. Movement of the HTD results in real-time updates to the portrayed texture to reflect the displacement. The end-to-end delay between displacement and update of the texture was measured to be 7.7 ms. However, we demonstrated that the theoretical limit of the end-to-end delay could be well below 1 ms, by making use of higher quality displacement sensor. Readily available heightmaps are used as a source of texture data. Because a method for characterizing the performance hitherto did not exist, a novel benchmark was developed that incorporates realistic textures and human participants. The results of our benchmarking yielded an average score of 57%. Since the device is the first of its kind, it cannot be compared to similar devices yet, but it can be used as a starting point on which future HTDs can improve.

CONTENTS

1

INTRODUCTION

1.1. TACTILE INTERNET

Since the advent of the internet, the amount of information exchange, such as audio, video, and data over long distances has seen a dramatic increase. And with faster internet reaching the remote corners in the world, it is paving the way for performing teleoperation, such as telesurgery, telerepair and maintenance over long distances. Furthermore, remotely executed tasks have numerous benefits, such as time-saving, money-saving, and convenience.

As is evident, the existing internet infrastructure supports audio and video communication in real-time. However, many tasks require the operator not just to have a visual and auditory display, but also get the perception of being present in the remote environment. It is clear that accomplishing a task by watching a computer screen is not the same as being in the place itself. And while it is currently possible to use virtual reality to be immersed in a virtual environment, this technique has not seen much use outside gaming and other specific niches. The virtual reality of today is missing a part of its "reality". And one of the main missing elements of current virtual environments the tactile information; the *sense of touch (haptic feedback)*. The haptic feedback is a major missing link in the chain that makes up virtual environments.

An important aspect of remote operations is the network itself. A fast network is needed to facilitate the proper execution of remote tasks. Relatively recent technology in the domain of teleoperations is *Tactile Internet*. Tactile internet is an internet network with adequate speed, reliability, and security to handle tactile data. Therefore, the tactile internet is defined as an internet network that combines ultra-low latency with extremely high availability, reliability, and security. It is difficult to achieve these stringent requirements with today's networks. Therefore, tasks involving tactile data are limited by today's internet.

Haptic feedback is the use of touch to communicate with the user. Haptic feedback is what is used to feel surface textures and forces while picking up objects. It is essential for interaction with the environment in order to perceive remote objects as physical and thereby perform accurate teleoperation.

1

The experience of touch comprises of two different types of feedback. These are kinaesthetic feedback (also often called force feedback) and cutaneous feedback (also often called tactile feedback). While kinaesthetic feedback is the experience of forces on muscles, cutaneous touch is the experience of pressure, temperature, vibration, and alike. Moreover, cutaneous feedback refers to statically contacting a surface. When there is active exploration over surfaces and objects, it is called haptic in haptic feedback, as that is the sensory system that textures are primarily experienced with.

One of the reasons that haptic feedback devices are not in widespread use is because the network specifications are stringent for haptic communication. This is shown in Table 1.1, adapted from [5], which shows general consensus for various Quality of Service parameters, namely audio, visual, graphics, and haptic data are stated, based on a number of studies.

QoS parameter	**Audio**	**Video**	**Graphics**	**Haptics**
Delay	150 ms	400 ms	[100 - 300] ms	[3 - 60] ms
Jitter	30 ms	30 ms	30 ms	[1 - 10] ms
Data Loss Rate	1 %	1 %	10 %	[0.01 - 10] %
Data Rate	[22 -200 kbps]	[2.5 -40 Mbps]	[45 -1.2 Mbps]	[128 kbps]

Table 1.1: QoS Parameters for audio, video, graphics and haptics [5].

The limitations in the current internet speed are the major reason that haptic feedback over the internet has not seen many practical implementations. But these limitations are decreasing due to faster internet by the day. It seems like in the near future, haptic data via the internet could be possible, and for limited applications, it may be already possible with existing internet networks. And haptic feedback over the internet has the potential to revolutionize remote operations in many segments of society, for example, healthcare, e-commerce, remote interaction, and more [2]. However, the practical implementations of this technique have not yet seen large scale use. For this reason, haptic feedback over the internet is becoming an increasingly popular research topic.

Tactile feedback is a part of haptic feedback, which is the interfacing of texture to the user. Haptic texture is a relatively new branch of haptic feedback systems, and therefore it has not seen much development. The technique is still in its infancy; currently, there are no devices on the market that interface the feeling of texture to the user. However, this technology has the potential to revolutionize some industries, for example, e-commerce, only a limited number of commercially available devices. These devices are based on vibration motors for displaying textures [19][14][17]. While this choice acts as a good starting point for the development of a more sophisticated device, it is insufficient to provide an immersive experience, owing to its low resolution and the display capability.

1.2. THESIS MOTIVATION

Fine-grained display of textures, such as that of silk, wood, etc, to the human user, plays a vital role in facilitating accurate recognition of textures of remote surfaces. Lack of high resolution and high-speed tactile devices in the market are a roadblock in our jour-

ney towards seamless haptic-enabled teleoperation. This motivates us to take up the problem of designing one ourselves. This is an extremely challenging problem since the mechanoreceptors on the skin are very sensitive, and they need to be stimulated with a fine-grained resolution. Hence, rather than designing a tactile device for the entire hand, we choose to design a fingertip-based tactile device.

1.3. PROPOSED SOLUTION

The contribution of this thesis is the design and development of a novel fingertip tactile device that we call HTD. This HTD is capable of interfacing haptic texture to a human fingertip. Our HTD follows the finger around over a planar surface and recreates the feeling of the texture dynamically when moving around. It has high enough resolution such that the fingertip cannot feel individual taxels. The actuation forces should be sufficient to feel surfaces clearly and to interface even highly uneven and bumpy textures. The response time of the device is fast enough to interface the surfaces correctly, even when moving around at high speeds. The textures can be loaded into the HTD using existing data structures for compatibility with existent textures and for easy visualization.

1.4. RESEARCH QUESTIONS

This report aims to design and present a novel haptic texture interface. In order to realize a design for a haptic texture interface, several research questions were identified. The main research problem statement is the following:

How can we develop a system which convincing and dynamically interfaces digitally-represented textures to the fingertip?

To answer the main research question comprehensively and to its full entirety, it is decomposed into smaller subquestions. These are the following:

1. **How can we achieve the right actuation to the fingertip?**
   To simulate texture, we need a way to recreate the same feeling as a real texture would to the fingertip. Therefore, a dynamic surface is needed, which can mimic the texture to be "displayed". We need a method that stimulates the senses in the fingertip. A more detailed elaboration is given in Section 2.1.

2. **What is the necessary resolution to convincingly simulate textures to humans, and how to achieve it?**
   The spacing between the discrete actuated pins must be small enough that the fingertip perceives the surface as continuous. Therefore, the discrete resolution must be better than what the fingertip can perceive. It could be compared to how a television screen shows pictures. An individual pixel does nothing like how images look. However, when enough pixels are placed close enough together, then it starts looking like an image. So the question is, from what resolution will individual points of feedback start to feel like one smooth surface. A more detailed elaboration is given in Section 2.2.

3. **How can we make the fingertip texture device dynamic, such that when the finger moves, the user experiences moving over the surface?**

The way that the fingertip perceives textures is mostly determined by what the fingertip feels when moving over it. It is difficult to recognize texture with the finger without it around. This can be shown by statically laying a finger on the surface, which makes it challenging to recognize the texture. It is thus essential for the perception of textures that the fingertip can move around over them. Therefore, the texture device must simulate the dynamics of the finger when moving over the surface. A more detailed elaboration is given in Section 2.3.

4. **How can different textures be stored, represented and rendered (displayed), in an accessible way?**
   We need a data format that can be used to store and exchange surfaces of texture digitally. This is used to upload textures to the haptic texture device and be compatible with existing technologies, which is elaborated in Section 2.4.

1.5. THESIS OUTLINE

To provide the reader with the necessary information about the research question and subquestions, Chapter 2 provides background information about the Thesis topic and states the requirements for our device. Then, in Section 3, related research works are introduced, and their contributions to the research questions are taken into account. In Chapter 4, the design and development of the HTD are elaborated, and design choices are elaborated. The subsections of that chapter answer the research subquestions. Then, in Chapter 5, the performance verification methodology is explained. The results of these tests are explained in Chapter 6. Then, the thesis will conclude with the conclusion, in Chapter 7.

2

THEORY

This chapter provides theoretical information required to properly understand the techniques and methods applied in this thesis properly. Further, a baseline for the specifications of the HTD is provided. Furthermore, the reasoning behind each research subquestion is explained. This chapter provides background knowledge for the design of an HTD.

2.1. ACTUATION

The first design challenge for the development of a haptic texture interface that needs addressing is the actuation. In an HTD, the fingertip is stimulated the same way as when it touches the real surface. Therefore, a way to mechanically imitate the surface texture is needed. Thus with actuation, there must be a way to recreate the feeling of texture to the fingertip mechanically.

Before the actuation system can be materialized, it is necessary to determine the requirements that must be met. For now, we will only discuss approximate figures since the exact figures are subjective and require empirical research. Many existing texture devices use 150 mN of force per pressure point for static force, in grids of 2.5 mm. This equals to 24 mN per mm^2. So we could state that one of the pin-grid actuation requirements would be that the actuation force must be at least 24 mN per mm^2, to be felt clearly. This output force must be adjustable, to follow gradients in the texture. A further requirement is that actuation is done with minimal latency and with high bandwidth, to follow the surface over which the finger moves. Lastly, since the skin of the finger dents in and out a little bit when pressure is applied, a minimum actuation stroke is required. The minimum required stroke is determined to be 2 mm since this is adequate to follow dents in the fingertip.

In order to generate this actuation, an actuator is needed that fits the requirements. Different actuator topologies could be used. Thus a topology must be chosen that fits our needs. Therefore, a variation of the most commonly used actuation methods is elaborated and compared in this section. Towards the end of this section, a summary of the advantages and disadvantages is given.

Eccentric Rotating Mass

A popular technique for realizing haptic feedback in tactile devices uses an Eccentric Rotating Mass (ERM) motor. These are simple Direct Current (DC) electric motors with an off-center mass on the rotating shaft, creating a vibrating sensation when driven. While the amplitude of the vibration is more or less constant, its frequency can be varied. These motors are already widely used in various devices, for example, smartphones and game controllers. Therefore, these ERM devices are widely available and cheap. A disadvantage of this design is that they need to accelerate and decelerate when a varying output is required, resulting in slow response time and low bandwidth than other actuation methods.

Electrotactille Display

An electrotactile display (also called electrocutaneous display) is a tactile feedback device with no moving parts but a grid of electrodes instead. These electrodes directly touch the user's skin and inject electrical pulses into the skin directly, creating the tactile feeling this way. Thus, an electrotactile display has no actuators, but it is added to this section since it replaces mechanical actuation. Since it has no moving parts, this type of tactile feedback has several advantages over other mechanical actuators, such as better mechanical robustness, low power consumption, low weight, and small size. The cost is also relatively low. However, the driver electronics are more complicated than many other solutions since the device generates a high output voltage and regulates the injected currents to keep the user from getting hurt.

Piezoelectric disks

Piezoelectric disks are actuators that use the piezoelectric effect to generate actuation. These actuators use a piezoelectric crystal to transform electrical signals into mechanical. A piezoelectric crystal is a material that deforms when an electrical voltage is applied over it. This deformation of the crystal is used to generate the required movement. This movement is generally tiny, in the micrometre range for most disk actuators. However, when these actuators are driven, a substantial mass vibrates at a high frequency, generating noticeable vibration. These piezoelectric crystals require relatively high voltage to get decent deformation, usually around 50 - 200V [7]. Therefore, relatively complex high voltage drivers are needed to generate the required high-voltage signals [20]. Further, since the amplitude of actuation is small, they are only meant to be used for vibrational or pulsed haptic feedback. Piezoelectric disks cannot be used with DC signals for continuous force outputs.

One of the limitations of piezoelectric disks is that they can only give pulsed and vibrational feedback, not a constant force. For a continuous movement using piezoelectrics, piezoelectric motors are made. A piezoelectric motor uses a vibrating piezoelectric crystal and transforms this vibration into linear motion [4]. The result is a very compact and lightweight linear motor. Due to the use of piezoelectrics, these motors still need complicated high voltage drivers, and the motors themselves are more expensive than standard electromagnetic motors.

Shape-Memory Alloy

Shape-memory Alloy (SMA) actuators use metal alloys that can be deformed when they are cold but return to their original form when heated. That is where the term "memory" comes from; the alloy remembers the pre-deformed shape when heated. The SMA is usually formed in strings that can be heated by current flow through the alloy. The current heats the alloy, thus reshaping the alloy to its pre-deformed state, which is how they actuate. Naturally, the heating and cooling of these SMA actuators cost some time, and therefore these actuators are known to have long response times and low bandwidth. On the other hand, these actuators are relatively compact, low cost, and relatively powerful for their size.

Electromagnetic actuation
Electromagnetic actuation is a topology where an electromagnetic coil attracts or repels another material. This material could be a magnet, electromagnet, or ferromagnetic material. The Lorentz force created by the electromagnet is used to generate the required force. Therefore, the output force (in steady-state) is proportional to the current through the coil, which is ideal for designing a constant-force actuator. Although, when used in continuous force mode, heat dissipation becomes the limiting factor in output force. The mechanical design of these actuators is relatively simple, making them cheap to buy and readily available. However, these actuators are generally bulkier than comparable piezoelectric counterparts.

Electromagnet-on-PCB
An electromagnet-on-PCB actuator is a variant of an electromagnetic actuator. The difference is that instead of an electromagnet wound of wire, the electromagnetic coil is incorporated on the Printed Circuit Board (PCB) itself. Having the coil on the PCB has the advantage that the coil does not have to be manually wound, can be very compact, and many coils can be placed close together. Further, the driver electronics can be directly incorporated on the same PCB, making the final design relatively compact. However, since a PCB is limited in the number of layers compared to a wound coil, the PCB coil generally has a lower output force and is more expensive since a high layer-count PCB is required.

Summary
Table 2.1 presents an overview of the upsides and downsides of the above actuators based on several performance aspects. This table will be used in the later sections to select the actuator topology for our HTD, once the required design specifications are determined.

2.2. RESOLUTION

The purpose of an HTD is to create the feeling of texture on the human skin. This feeling can be recreated when the fingertip's skin is stimulated the same as it with the real texture would. There are ways to inject electrical pulses into the skin directly, for example, in [9], by a so-called electro-tactile display. However, an electro-tactile display will not be used in this thesis, as electrically stimulating the skin does not result in a realistic feeling

Type	Max Force	Variable Force	Speed	Size	Cost	Driver Simplicity
ERM	+ +	+/-	-	-	+	+ +
Piezo Disk	+	-	+	+ +	+	- -
Piezo Motor	+	-	+/-	+ +	- -	- -
SMA	+	- -	- -	+/-	+/-	+
EM on PCB	+/-	+	+	-	-	+
EM actuation	+	+	+	-	+/-	+

Table 2.1: Actuator topology comparison chart, showing the global characteristics of different actuator types.

of surface texture. There is a more elaborate explanation on this topic in Section 3.2. For a more realistic feeling, mechanical stimulation on the fingertip can be used, which dynamically mimics the surface texture as the finger moves over it. Therefore, a way to mechanically imitate the surface texture is needed.

Figure 2.1: Example of a Pin-art, as advertized in an online webshop[1].

However, replicating a surface texture by automating a Pin-art requires an enormous number of actuators since every pin needs to be individually controlled. The number of required actuators is a function of the covered surface area and resolution. This results in a bulky and costly device. A sensible way to decrease the necessary number of mechanically actuated pins is only to have pins that actually touch the fingertip and follow the fingertip with them. Thus the whole pin-grid moves together with the finger. It follows the finger around and dynamically recreates the texture under the finger. This could be

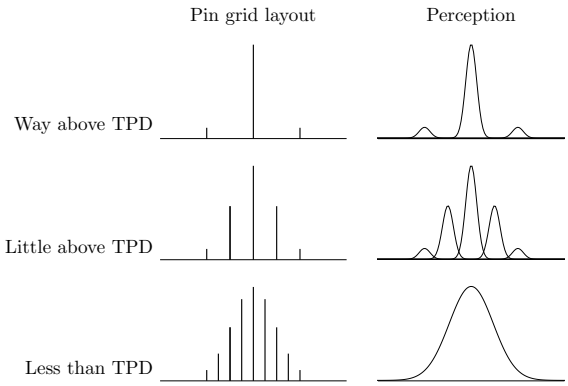


Figure 2.2: Visualization of the two point discrimination

a good starting point for a mechanical surface texture interface.

For the design of a shape display made up of electrically movable pins, an important aspect to consider is the resolution of the display. With resolution is meant, how closely the pins must be spaced out, and what size they have to be. This idea relies on the following technique. When a grid of pins movable pins is used, the pins imitate the higher and lower spots of the texture. And when enough pins close enough together are used, the skin is no longer able to distinguish the individual pins. It is perceived as one smooth surface, instead of individual pins. This technique is analogous to the workings of a television screen. If a television screen were to be built with pixels of a few millimetres in size, the visuals would not look convincing because every individual pixel is visible. But when enough small pixels are spaced close enough together, we cannot distinguish the, and the grid of pixels is perceived as one smooth picture. The same is true for this shape display. When the pins are small enough and close enough together, it is perceived as one smooth surface, and the distance between the pins is called the Two Point Discrimination (TPD). This is visually shown in Figure 2.2.

However, to take advantage of the indistinguishability of small pins spaced closely together, it is first necessary to first figure out what pin spacing is adequate. In other words, what pin-grid resolution is needed for it to be perceived as a continuous surface. Therefore, a short literature review was performed.

It turns out, the concept of indistinguishability of pins spaced closely together is called the TPD. The TPD is the minimum distance where the receptors in the skin can still discriminate between the two spatially separate points. According to [28], the TPD differs from where on the skin the test is performed. This is expected since human skin is not equally sensitive everywhere. For our HTD design, the interest lies in the sensitivity of the fingertip. The resolution of the HTD must be better than the HTD of the fingertip.

Sang-Yeun won et al. [28] performed various measurements of TPD on many places on the skin, including the fingertip. From their results, the mean HTD on the fingertip is approximately 2.3 mm. This implies that using a grid spacing of 2.3 mm, for 50% of the users, this is adequate. For this reason, it was decided that the maximum grid spacing

requirement is 2.3 mm, but smaller spacing is preferable if possible.

When a particular grid spacing and number is decided, there also is the challenge of how actuators fit within the distance between actuators. It could be that decent actuators would not fit within the grid spacing, as mentioned in 2.1. If this is the case, another solution must be found to resolve this dense grid actuation.

2.3. DYNAMICS

In Section 2.1, it was stated that a sensible way to decrease the number of actuated pins on a shape display for the fingertip is by only having the pins that actually touch the fingertip itself. It follows the finger around and dynamically recreates the texture under the fingertip. However, this dynamic recreation of texture under the finger poses another design challenge. The texture device now needs to keep track of the location of the finger. At what location on the texture is the fingertip at any given moment. That is what is meant by dynamics in this context. The dynamics of the fingertip HTD, to be able to follow the movement and adjust the output to the fingertip accordingly. Since the finger must be free to move around over the surface, the freedom of the device spans a two-dimensional surface; it has 2-Degrees of Freedom (DOF). Now, this location tracking system could pose several dilemmas concerning latency, precision and update rate.

An often-heard ballpark figure in the tactile internet is that for haptic data, a maximum end-to-end delay of less than 1 ms is required[24]. Since our HTD will work locally, the end-to-end delay is only dependent on the latency of the HTD itself. However, it is still a very stringent requirement. It impacts the following parts of the design. All constituents of HTD, such as location detection, actuator and driver, must be extremely responsive.

Further, more and more studies indicate that depending on the specific task to be performed, more latency could still be acceptable. So to make sure the design requirements are not too stringent, and to make sure the final design will not only be limited by this requirement, it was decided that for our HTD design, a latency of up to 10 ms will be considered acceptable. Later empirical tests can demonstrate for what specific tasks this latency will be noticed, and for which tasks it is adequate. Further, the addition of dynamics to the HTD poses more design challenges. For example, when moving around, the location must be updated with a sufficient update rate and enough accuracy to convince the user that it moves its finger over the texture. Moreover, the location detection mechanism is not allowed to limit the free movement of the fingertip.

2.4. COMPATIBILITY

A practical but essential aspect of the development of a HTD is the ability to digitally diverse textures. We must be able to create and store textures on the HTD. The digitally stored texture is already used in other sectors, especially in the gaming industry. Textures are used for the visual representation of surfaces. What is needed is a solution capable of using these existent textures so that the HTD can be used with preexisting texture data. Then this texture data can be uploaded to the memory texture interface, such that when it is used, the texture data is stored locally and can be read at high speed, so latency is minimized. An optimal solution would be when the existing textures could be used and

be compatible with the developed texture interface.

2.5. SYSTEM REQUIREMENTS

This chapter concludes with a list of all the system requirements for the HTD. These requirements come from the previous sections of this chapter, where every individual requirement is stated. The final HTD design should comply with these specifications, which form the foundation for the HTD design. These system requirements are shown in Table 2.2.

It must be stated that the numbers in this table are guideline figures. For some of the requirements (for example, the numbers for the required actuation force), the exact requirements must be determined by empirical tests. Therefore, these requirements could be seen as ballpark figures. The exact numbers for some of the specifications depend on test results, which will be done in Chapter 4.

Specification	**Minimum**	**Recommended**	**Maximum**	**Unit**
Grid Spacing	-	2.0	2.3	mm
Actuation Force	24	50	100	mN per mm^2
Actuation Stroke	2.0	2.0		mN
Response Time	-	1.0	10	ms
DOF	2	6		-

Table 2.2: System requirements

Further must-haves that are not part of the specifications are that the device uses existing texture data types and that the actuators deliver variable output force depending on the texture. These requirements are used for comparing the related works to the work that is desired. It can be determined what design elements can also be used for our HTD. This comparison to related works is made in Chapter 3.

3

RELATED WORKS

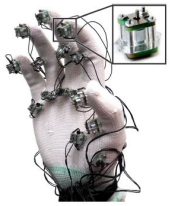
A literature review was performed to get an overview of related research works that could be relevant to this HTD design. Design considerations and findings of these prior researches might be useful in this work. From this literature review, a variation of the for this Thesis relevant research works are listed in this section.

There are many related haptic feedback devices made in the past years. The literature review covers devices that are related, but not necessarily designed for realistic texture feedback. This is because the field of haptic texture is relatively new, and not many devices are developed for this purpose. Further, there are numerous related devices, and it was tried to showcase a selection of different relevant techniques and form factors in this section. It must be stated that many more devices and papers are reviewed as part of this literature review, but there is no point in covering them all.

In this chapter, it was tried to compile some different designs of achieving the haptic feedback. This was done to give a more broad view of what related works are already done and to show what is possible. A short description of the functional principles of the design is given, and in the end, an overview is given to compare the specifications for every reviewed device. The most important aspects of our design are, as described in Section 1, the following: Actuation, Resolution, Dynamics, and Compatibility.

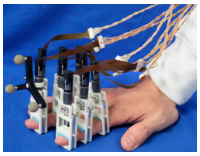
3.1. GLOVE HAPTIC FEEDBACK DEVICES

Many haptic feedback devices use a glove-like design, with actuators in various places on the glove, for the haptic feedback. These are mostly used for virtual reality purposes and designed to impose minimal limitations on the user's free movement. For example, Tactiles by Vechev et al. [25], who designed with electromagnetic actuators for rendering continuous contact and spatial haptic patterns in virtual reality. A total of 15 electromagnetic actuators, placed in various places on the hand, are used for the tactile sensation. The actuators consist of a bi-stable electromagnetic latching mechanism. A magnet can move between both latching positions via pulses of electrical current through a coil on a PCB. They used 8-layer PCBs to get the number of turns on the coil to achieve actuation between the two states. The exact specifications are shown in Table 3.1.



(a) Tactilles

(b) Vibrotactile Glove



(c) Micro surface undulation display

Figure 3.1: Glove-like haptic feedback devices

Further, the vibrotactile glove designed by Martínez et al. [15] used vibrotactile actuators at variable frequencies, which gives the user the sensation of touching and enclosing virtual objects with their hand. For the actuation, they used dc electric motors, which spin an eccentric mass around. This gives a vibrating sensation to the user. Tanaka et al. [21] developed a glove-like system that generates a pressure feeling to the whole finger using a wearable haptic display. Small motors rotate a ball screw, to induce pressure at three places on the finger. Images of these devices are shown in Figure 3.1.

3.2. SOLID STATE HAPTIC DEVICES

With solid-state haptic devices is meant the devices that create tactile feedback, with no moving parts. For example:

HamsaTouch
The HamsaTouch device is originally designed to give visually impaired people the ability to get an image of the environment using tactile feedback. The camera of a smartphone creates the image, and HamsaTouch translates this to tactile feedback using a so-called electro-tactile display. An electro-tactile (also called electrocutaneous) display is a type of tactile display, with no moving parts, which uses electrical current to stimulate the nerves in the skin directly. An example of such a display is shown in Figure 3.2. This method of tactile feedback has, therefore, no moving parts. Instead, it uses an array of small electrodes that supply the electrical current into the skin. Therefore, this type of display has several advantages over other mechanical style tactile displays, such as better mechanical robustness, low power consumption, low weight, and low thickness.

Due to their direct electronic stimulation, electro-tactile displays have disadvantages over a mechanical tactile display, according to "Electro-tactile Display: Principle and Hardware" [10]: The strength of the sensation is not stable because the conductivity from the contacts to the skin is relatively finicky. The use of a conductive gel and the use of electrodes that do not corrode are techniques to help but do not solve this problem completely. Further, because currents are directly injected into the skin, usage can be experienced as painful. Moreover, there is no method for rendering natural tactile information. Due to the injection of currents into the skin, it creates a feeling in the skin, but this is not the same as when the skin is stimulated naturally. This last point is the crucial downside of the electro-tactile display. One of the main design criteria is that

Figure 3.2: Example usage of an electro-tactile display (HamsaTouch), by Kajimoto [11]

the system can convincingly interface texture to the fingertip. Thus, it is a big dilemma when the tactile display used to do this cannot replicate a natural feeling in the fingertip. Therefore, it is decided that this type of tactile display will not be considered for the design of the fingertip texture interface.

3.3. PIN-ARRAY HAPTIC DEVICES

Devices using a pin-array structure can more closely recreate the feeling of haptic texture to the fingertip than vibrational or electrotactile displays. Therefore, these are potentially more relevant then the glove haptic feedback devices, and thus are covered here in more detail.

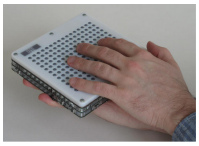
BlindPAD

The first related work reviewed here is the BlindPAD [29], which was designed for visually impaired people. They state that BlindPAD is developed to give visually impaired people veridical touch-based information, exploiting and enhancing their residual sensory abilities.

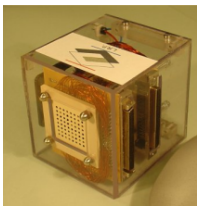
The design consists of a 12 x 16 taxel array, where every taxel can be individually controlled to an up-or a downstate. Every taxel has only these two discrete states, where both states are latching. For this actuation, electromagnetic actuators are used. According to their project report [3], they also tested shape-memory polymer technology but adopted electromagnetic actuators because it granted high forces, high displacement, and high reliability, from their research. The actuators consist of a magnet that can move between two printed circuit boards (PCBs). The magnets can be pushed or pulled by two electromagnetic coils printed directly on the PCB traces. They used two 6-layer PCBs to get the required number turns on the coils in the limited surface area under the taxels.

Compact Tactile Display

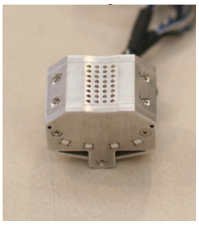
The Compact Tactile Display for the Blind [26] was designed as a novel tactile display device with advantageous features for the blind. The design consists of a grid of 8 x 8 tactile pins, actuated by shape memory alloys (SMAs). Their paper states that commercially available tactile devices are piezoelectric refreshable Braille displays, have a



(a) Blindpad



(b) Compact Tactile Display



(c) SaLT

Figure 3.3: Braille-like shape displays

way higher cost than their design (between 7500 and 15000 USD). However, to the burdensome piezo-actuator module that drives the pin, they are not suitable for graphical information display and bulkier than their design. Further, they state that since they use alternative technologies like SMAs, their concept intends to make possible compact highly-portable text-graphic Braille-like displays for 200 USD or less. The potential applications include virtual reality, sensory augmentation, sensory substitution, robotics, game, and entertainment, among many others.

SaLT

The Small and Lightweight Tactile Display (SaLT) [12], aimed to build a small and lightweight tactile display system. The system, including the display modules and complete controller parts, will be wearable by a user. The mechanical design consists of a 4 x 4 actuator array, using piezoelectric actuators. Each actuator drives two pins simultaneously; therefore, the pin array is 8 x 4 pins. The actuators used for the design are piezoelectric actuators, so-called TULAs (Tiny Ultrasonic Linear Actuator). These are subminiature linear ultrasonic motors using the piezoelectric effect for linear movement. These TULA motors are also often seen in-camera autofocus systems, medical equipment, and other applications where small size is critical.

Lastly, what is not stated in the paper, these displays are relatively expensive since the specific TULA actuators used in this display cost $34.00 each when a minimum quantity of 50 is bought (with lower quantities, the price is higher, in a quantity of 5 they cost $70.00 each). This brings the cost of only the actuators for this display to $34 * 16 = $544.00.

3.4. SPECIFICATIONS AND ANALYSIS

A summary of the specification of the various mentioned haptic devices in recent research works is presented. This data can be seen as the current state of the art in haptic feedback devices. It must be stated that not every detail of every device is shown in the table. As not every paper states the same details of their design. Table 3.1 shows the technical specifications. These technical specifications are more of interest for this HTD design, as these are the same parameters as the requirements of our HTD design.

This table views to what extent the existing state of the art devices meet the requirements for our HTD device. This information can be used to determine what design deci-

Name	Taxels	Resp.	BW	Taxel force	Pitch	Travel
Blindpad	192	10 mS	0.5 Hz	200 mN	8 mm	0.8 mm
Compact Tactile Display	64	-	1.5 Hz	320 mN	2.6 mm	1.0 mm
SaLT	16	-	20 Hz	196 mN	1.5 mm	0.5 mm
Tactiles	15	50 mS	200 Hz	200 mN	-	2.0 mm
Vibrotactile Glove	11	150 mS	-	-	-	-

Table 3.1: Technological specifications of related works

sions could also be taken for our HTD design. Also, it can be quickly shown that there is not a single device that meets all of the requirements for our HTD design, and thus that there is the necessity to develop our own.

4

DESIGN METHODOLOGY

This chapter explains the methodological steps that we took in our iterative design process. This design process consists of numerous sub-challenges that individually need to be examined and resolved. Many parts of the design were tested, characterized, and modified, following this iterative design process. This process applies to many design decisions, often requiring several design iterations before the result is as desired. In this chapter, these design steps are explained, the relevant iterations of the process are elaborated, and the research questions, as stated in Section 1.4 are answered.

4.1. ACTUATION

Actuation is one of the primary design aspects for the development of the HTD. The actuator design also affects the resolution and dynamics (latency) and form factor of our HTD. Therefore, the actuation method is the first topic that is examined. The design requirements specific for the actuation are adopted from Chapter 2 and stated in Table 4.1.

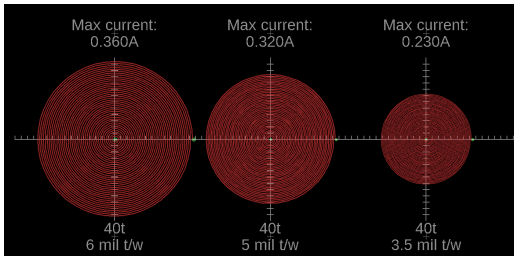
The construction of the actuators is largely determined by the design criteria of Table 4.1. This list of requirements is compared with Table 2.1, which shows the general qualities of the different topologies. The first four columns of this table are of most interest to us. Consulting this table shows that the electromagnet on PCB and the discrete electromagnetic actuator perform best for these specifications.

4.1.1. ELECTROMAGNET-ON-PCB ACTUATOR

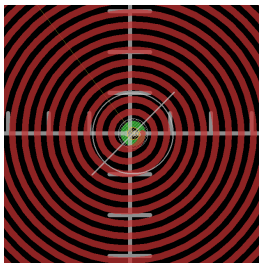
An electromagnet-on-PCB has a few advantages over the wound coil design, but the output force is lower per area; therefore, we want to verify whether that is a feasible solution. The advantage of a coil-on-PCB is that it eliminates the need for manual winding, thus provides consistent coils while also providing room for customization of the actuator. The downside of this actuator topology compared to a discrete electromagnetic actuator is that the output force is lower for a given area because the output force of an electromagnet is a function of current and number of windings. For this reason, it does not matter whether we use thick or thin traces, which we will explain using the following

Specification	Minimum	Recommended	Maximum	Unit
Actuation Force	150	200	500	mN
Actuation Stroke	2.0	2.0		mN
Response Time	-	1.0	10	ms

Table 4.1: Actuator requirements



(a) Coil Designs



(b) Coil design close-up

Figure 4.1: Electromagnet on PCB design

example. Let us say we use wide PCB traces, the current can be higher due to the low resistance of the traces, but the number of windings per area is low. In comparison, when half as thick traces are used, twice the number of windings will fit within the area, but these traces are only capable of handling half the current, resulting in the same output force. To conclude, the output force is only influenced by the coil area and the number of layers in the PCB. Thus, we need to verify whether the electromagnet-on-PCB actuation yields an acceptable output force, and thus is a feasible option.

To determine whether an electromagnet-on-PCB actuator is a feasible solution, we designed a few coils according to the design rules of the PCB manufacturer [8]. These coil designs are shown in Figure 4.1, where the top layer copper traces are in red, and the white axes show one-millimetre intervals. A zoomed-in version of the same is shown in Figure 4.1a for demonstration, but the coils we do the calculation on have a diameter of 5 mm. Since our maximum desired grid spacing is 2.3 mm, the necessary output force is 150 mN. Using this design, we determined the output force of these coil designs and verified whether these actuators could generate sufficient force.

The output force of the actuator can be determined using the Gilbert model. The Gilbert model assumes that magnetic charges reside near the magnetic poles, which result in the magnetic force between the magnets. This model is used because the calculated forces are accurate, even when the magnets are close together. The following formulae from the Gilbert model are consulted to calculate the output force[27]:

$$B = \mu_0 \frac{NI}{L} \text{ and } M = \frac{B_0}{0.5\mu_0} \text{ and } m = MV \tag{4.1}$$

$$F \approx \frac{3\mu_0}{2\pi} m_1 m_2 \frac{1}{x^4} \tag{4.2}$$

The magnet and coil used for this calculation have the following specifications:

Board	Force	Order Cost
2-Layer	26 mN	$39
4-Layer	52 mN	$92
6-Layer	78 mN	$200
8-Layer	104 mN	$500
10-Layer	130 mN	$586
12-Layer	151 mN	$793

Table 4.2: Multi-Layer board comparison

- magnet:
  - B =1.2 T
  - l =0.01 m
  - r =0.0025 m
- Coil-on-PCB:
  - I =0.32 A
  - N =9 turns per layer
  - r =0.0025 m
  - Distance between coil and magnet: 5 mm

$$m_{\text{magnet}} = \frac{B}{\mu_0/2} \times V \approx 0.12\,\text{A/m}^2 \tag{4.3}$$

$$m_{\text{coil}} = \frac{\mu_0 \frac{NI}{l}}{\mu_0/2} \approx 1.77 \times 10^{-4}\,\text{A/m}^2 \text{ per layer} \tag{4.4}$$

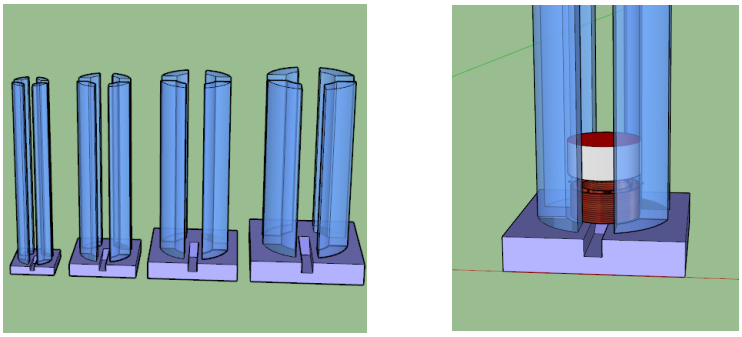
The force for a single layer coil-on-PCB can now be calculated using:

$$F \approx \frac{3\mu_0}{2\pi} m_1 m_2 \left(\frac{1}{x^4}\right) \approx 13\,\text{mN} \tag{4.5}$$

We performed the same calculation for boards with more layers, and it becomes clear that this topology does not yield the output force we desire. The calculated output force of various multilayer boards and their associated cost[1] is shown in Table 4.2. Twelve-layer boards are needed, to reach the minimum requirement for the output force specification, leaving no headroom. We desire a bit more headroom for the actuator force.

The strongest magnets (N52 grade neodymium) on the market were used for these experiments, combined with the absolute maximum current through traces of this thickness. It is clear that this topology is not suitable for our purposes; wound coil actuators are the better option here. Although wound coil actuators are higher in size, their footprint is considerably smaller than coil-on-PCB actuators. Moreover, since we desire to have certainty that the output force is sufficient for adequate actuation, we rather over-engineer the actuators to be bulkier but more powerful to be sure we have ample output

[1] price values based on 50 x 50 millimeter PCBs by PCBWay, on 20 October 2020



(a) Different footprint designs (b) Coil under the magnet

Figure 4.2: Electromagnetic actuator design

force. To conclude, we decided that a bulkier wire-wound coil is a better solution than a coil-on-PCB.

4.1.2. DISCRETE ELECTROMAGNETIC ACTUATOR

We tried to find off-the-shelf solutions for our electromagnetic actuator requirement. However, it turns out that the stringent response-time requirement in combination with an adjustable constant force output yields an actuator specification for which no solution is available on the market. Therefore, we need to design and develop our own actuators.

For the actuator design, Google SketchUp is used due to its compatibility with most 3D printers. The designs can be 3D printed on a Fused Filament Fabrication (FFF) style 3D printer (the Creality CR-10s). Coils are wound by hand, using 0.4mm enamelled copper wire.

In our first actuator design, we chose for a stationary coil, exerting force on the magnet. It does not matter whether the magnet or the coil is the moving part for the output force. Further, the wound coil can be used to attract or repel the magnet (or both when using two coils). For this actuator design, we require an adjustable output force. However, when using magnetic attraction, the pulling force using a constant field increases with the square of the distance. This makes it impossible to deliver constant force without the use of a feedback system, requiring additional space. The development of a feedback system for every actuator to deliver constant force is deemed unnecessarily complicated, and magnetic repulsion is used instead. The precise graph of the output force versus distance will be explained in more detail in Section 4.1.3. Lastly, since these actuators will be used in a grid, we require relatively small footprints. Since it was not definitive whether these small footprint actuators would exert sufficient output force as per the specifications, it was decided that a few different sizes of actuators be made per actuator design.

The first prototype was designed according to these specifications and is shown in Figure 4.3. As shown in the image, the shaft where the magnet moves in consists of four separate rounded pieces. The intention behind this design is that the inner workings of

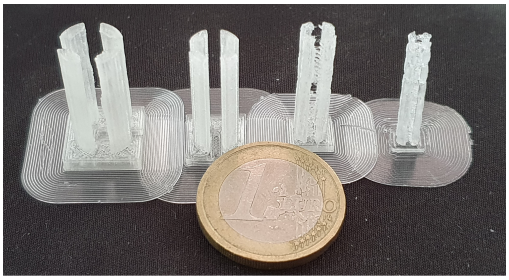
the actuator were still visible when built. For this same reason, the actuator was designed for translucent Polylactic Acid (PLA) filament, which makes the operation extra clear for demonstrational operation. Further, the coil is wound beforehand and placed on the bottom of the actuator with the wires coming out of the notch in the design (Figure 4.3). Now, the coil can repel the magnet, forcing it upwards through the shaft.

This type of actuator was printed and built, but no detailed measurements were performed because of underwhelming results. Firstly, the split-shaft turned out to be troublesome during printing and fragile to handle if the printing succeeded. Further, since we are using a FFF printer for printing the actuator, the small 4-piece shaft was printed roughly. This roughness was caused by so-called "zits", small blobs on the 3D-printed surface. These zits occur when the printhead starts and stops moving over the model, melting a blob in the model when standing still [16]. Often these zits can be resolved by reducing the number of times that the printhead needs to stall. Furthermore, stringing was a problem in these small splits. Stringing happens when printing separate straight pieces closely together. It originates from the 3D printer nozzle, which leaks some molten plastic when it should stop extruding plastic. When the printhead then moves around, it creates a string of plastic between the close together pieces.

Lastly, in one of the early prototype prints, the print was mistakingly started without a brim around it. This resulted in the actuator print of Figure 4.3b, which looked acceptable on the bottom and destroyed from about 2 mm and up. The printer had a difficult time printing this design. This is the result of a print that came loose of the build plate during printing. A brim would have resolved this problem entirely. A brim around a print is an extra border of plastic material extending along the bottom of a printed model, as is visible around the bottoms of the actuators in 4.3a. This is done to improve bed adhesion. When 3D printing is started, the slicer software provides four different bed adhesion options. Firstly, a raft can be chosen. A raft is a large lattice that is printed around and under the model. It improves bed adhesion and prevents warping of the model during cooldown. Secondly, a brim can be chosen, which is used for our actuator prints. A brim is a border around the model, which does not extend under it. A brim also improves bed adhesion but wastes less material and yields a cleaner bottom of the part, though it does not prevent warping. Warping is only a problem for designs with larger footprints (>10 cm2), which is not the case for these actuator designs. Thirdly, a skirt can be chosen, which does not improve warping or bed adhesion. It is only used to prime the extruder and establish smooth flow before starting on the model. Lastly, "no adhesion" is also an option. This option is generally not advised but saves material since no extra extrusion is needed.

The split-shaft design had to be abandoned to resolve the problems resulting from the small structures in the design. We developed a new design that works similarly but uses a single shaft with one opening on the side. This single opening eliminates the small straight guide pieces but keeps the visibility to the actuator's inner parts. This design eliminates both the stringing problem and the zits from the 3D printing process, compared to the previous design. Another advantage is that now the magnet can move with less friction to the actuator walls since these are smoother since they consist of a continuous segment.

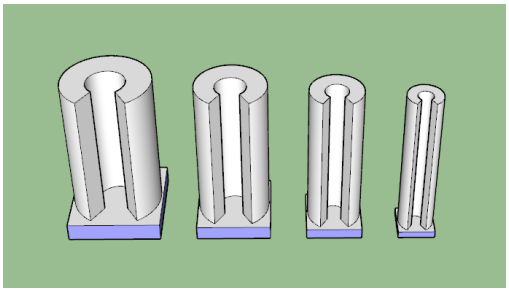
The problem with this design turned out to be that the coil is placed under the mag-



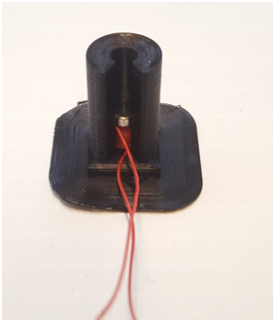
(a) 3D printed actuator shells

(b) No-brim failure

Figure 4.3: First 3D printing results



(a) Opened shaft design

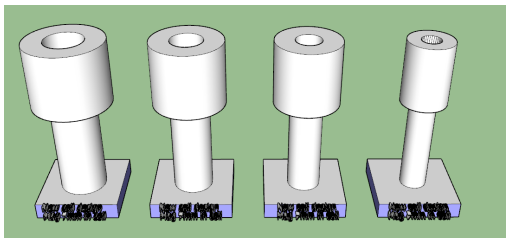


(b) Open shaft prototype

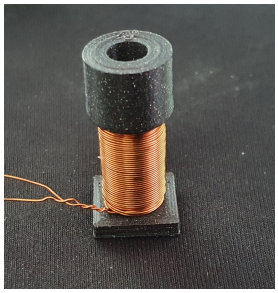
Figure 4.4: Actuator prototype

net, and there is no space for a large enough coil to fit. This small coil severely limits the output force of the actuator. Therefore, this design was replaced with a new design that encloses the magnet with a large coil. In this way, a large coil can be wound, the volume of the actuator is used more efficiently, and the output force can be larger. A recessed section in the cylinder is designed for this larger coil, where the coil is wound around. This also helps with the consistency of these actuators, when many of them are wound. The disadvantage of this design is that the visibility to the actuator's inner parts is removed.

First testing showed that these actuators were a success, but that there was however a problem with the PLA which these actuators are printed from. The problem is that when the actuators run at maximum output force for a prolonged time, they heat up considerably due to the DC-resistance of the coil, and the temperature reaches the glass-transition temperature of the PLA which is 55 °C[18]. If this happens, the actuator shaft deforms, obstructing the magnet's movement and effectively breaking the actuator. This problem was resolved by changing the printing filament to Acrylonitrile Butadiene Styrene (ABS) plastic filament. ABS has a glass transition temperature of 105 °C, giving us significantly more headroom. Despite that, the self-heating of the actuator is still what is currently limiting the output force of the actuators. We could increase the output force further by adding more cooling capability, for example, by adding forced air convection,



(a) Opened shaft design



(b) Closed shaft prototype

Figure 4.5: Actuator prototype

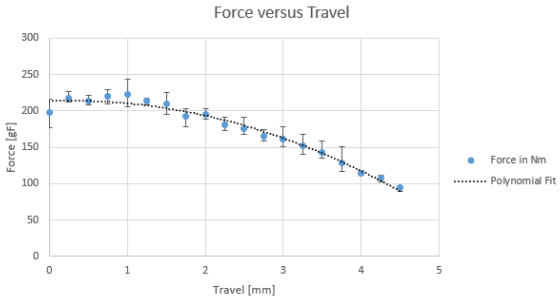
but this was deemed unnecessary since the output force is already ample.

4.1.3. RESULTS

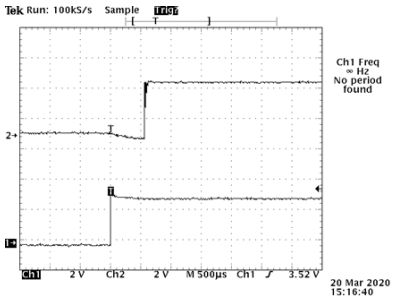
To evaluate and characterize the performance of the developed actuator, we measured various parameters. We started with the evaluation of the output force. The actuation force was measured over the full motion, to characterize the force over the whole output range. This results in a force versus distance, shown in Figure 4.6a. According to the equation 4.2, we expected the curve of the output force to follow a polynomial curve. Therefore, a polynomial-fit trendline was placed over the figure, shown in black. What is immediately quite clear is that the output force between a stroke of 0.5 mm and 2 mm is relatively constant. This is perfect for our application since the actuation stroke is always within this range. This relatively flat part of the curve can be used in our advantage since it means that a constant output force can be realized without the use of a feedback system

Measurements of the response time were also performed. The response time was measured using the following method. The actuation pin was connected using a thin wire connected to the +3.3 V, and above the actuator pin, we placed a strip of aluminium foil, connected to the oscilloscope probe. When the actuator moves up, it touches the aluminium foil, thus connects the +3.3 V rail to the oscilloscope Channel. To measure the time difference between turn-on and movement, Channel 1 of the oscilloscope was connected to the actuator input, and channel 2 thus to the foil. We can now measure the actuator's response time by measuring the difference between the input going high, and the output goes high. The result of this measurement is shown in Figure 4.6b. Note that the signal on Channel 2 is floating around a little bit before the rising edge. This is because the probe is not connected to anything before contact, thus picks up ripple and noise from the environment. However, this does not affect the measurement outcome since the amplitude is way higher than that of noise. From this measurement, we can see that the actuator's response time is just below 0.6 ms. This is an extremely desirable performance for tactile internet applications, leaving the headroom for delays in the other system parts.

Finally, we measure the relationship between the input setpoint and the output force. This is important to be determined since it helps us understand how the input is to be



(a) Actuator force against the stroke



(b) Response time measurement

Figure 4.6: Actuator measurements

4

adjusted to get a specified output force. We desire a linear relationship between input and output; otherwise setting a precise output force is difficult. Since we are operating the actuator at constant output force (steady-state), the actuator is used as a resistance, where the current through this resistance determines the output force. Since that actuator should behave as a constant resistance, we adjusted the actuator input current by a Pulse Width Modulation (PWM) output of the microcontroller, and an L298 driver Integrated Circuit (IC). The measurement of setpoint against real output current is shown in Figure 4.7.

From this figure, it is evident that the output force did not scale linearly with the setpoint. It was soon discovered that it is incorrect to model the actuator as a constant resistance when driving it with a PWM signal. Since the actuator is an inductor, when not driven at DC, the impedance is frequency dependent. A correct model of the actuator would be a resistance in series with an inductance. Since PWM is an Alternating Current (AC) signal, the actuator's inductivity decreases the current in the middle of the actuation range. The inductor resists quick changes in current, therefore at a 50% duty cycle, less than the set 50% of the current flows through the actuator. It is possible to program this curve in the microcontroller and compensate for it in software; however, this creates more driving overhead and relatively slow response time, since the inductance asymptotically reaches the desired current. A better way would be using a Constant Current (CC) driver circuit, which adjusts the voltage over the actuator, and keeps the current precisely at the setpoint. This ensures that the correct output current is reached with minimum delay. Therefore, another schematic was designed that utilizes constant current output drivers. One disadvantage of such a system is that the constant current drive is less efficient since it drops the voltage to achieve the correct current. However, since the output force setpoint is highly important for our design, this was the chosen solution to achieve a linear input to output relation. The result of this CC driver is shown in Figure 4.7.

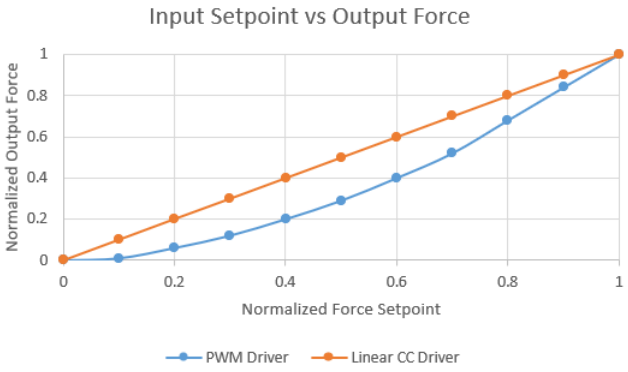


Figure 4.7: Actuator Linearity

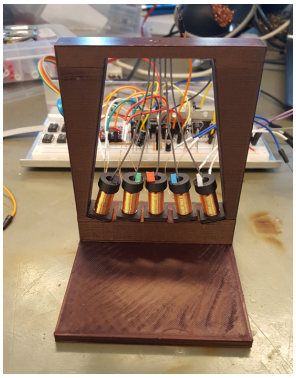
4.2. RESOLUTION

The resolution of the HTD largely determines the performance; therefore, a well-evaluated decision must be made on how to achieve the optimum resolution at the fingertip. As explained in Section 2.2, we require a minimum resolution of 2.3 mm, determined by the TPD of the fingertip of the human hand. Besides, the self-designed actuators are wider than that spacing. Therefore a system is needed to guide the actuator force to the small surface area on the fingertip. Moreover, the permanent magnets inside the actuators interfere with each other when placed near together. It was found that we need 15 mm actuator spacing to remove this effect almost entirely. A solution is the use of pushrods, which create a distance between the actuators and the fingertip. This way, the distance between actuators can be increased while all pushrods come together and form a high-resolution gird at the fingertip. This way, the interval between the actuated pins can be way smaller than the distance between the pins at the fingertip. Another advantage of the use of pushrods is that we can now adjust the resolution of the grid, just by adjusting the relative position of the pushrods. This sounds like a great idea, but it brings some design challenges with it, such as are these pushrods made, how are they guided to a dense grid on the fingertip with minimum friction.

4.2.1. PIN-GRID DEVELOPMENT

Lets first address the creation of the holes that guide the rods to the fingertip. The holes must be spaced closer than 2.3 mm, and if possible even at 2.0 mm intervals. One way is to use a small enough drill, and drill holes at precise intervals. This was tried in the first prototype. To test whether this pushrod pin-grid idea works as intended, we developed a small scale HTD. This small prototype can also give us insights to pitfalls which can be resolved before we scale up to the final design. At first, it was tried to drill the holes in an angled piece where the finger can rest in, which is shown in Figure 4.8a. The small-scale HTD is shown in Figure 4.8b. However, as is shown in the image, the drilled holes in the angled piece were no success, so the prototype used a flat plate to rest the finger on instead.

(a) The finger plate with drilled holes. The holes don't have the required accuracy since they are drilled manually.



(b) Prototype build with a straight fingerplate instead, since drilling the holes in an angled piece was infeasible.

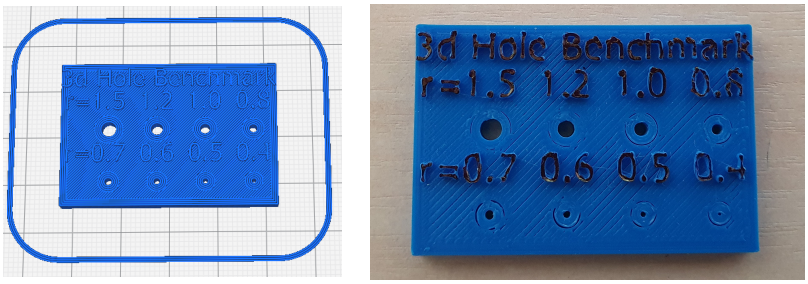
Figure 4.8: Grid prototype

The pushrods are made out of iron wire, which turned out to work well. The advantage of using iron wire is that it is ferromagnetic and therefore sticks to the magnets in the actuators. This is advantageous since it makes assembling and disassembling the actuators easy. Also, it saves the step of fixing the rods rigidly to the magnet in the building process. Further, the top of the pushrod can be relatively easily shaped to be the required shape, using a Dremel with a grinding wheel. Therefore we decided to use these iron wire pushrods for the HTD designs.

The manually drilled holes, however, were no success. It is hard to drill at the precise intervals, since the drill runs away over the plastic, and sticks to the plastic during drilling due to the self-heating of the drill. What makes it more difficult is that the drilled holes need to be at precise angles, such that the pushrods reach the actuators below. The result of the manually drilled holes is shown in Figure 4.8a. Even on a flat surface, the drill runs away and does not drill straight. Soon it was decided that manually drilling the holes for this one-dimensional grid is hard and that this will not work for the two-dimensional pin grid. Another solution for creating the pin-grid was needed.

Since many of our parts are already created using the 3D printer, we wanted to evaluate whether it would be possible to 3D print the grid of holes for the pushrods. This would save the manual and imprecise drilling process and the creation of much more consistently made grid. Further, we can now specify precise angles and grid spacing of the pushrods. However, printing holes with a diameter of 1.0 mm and intervals of 2.0 mm and at precise angles is going to be a challenge our 3D printer. Therefore, we need to investigate whether we can reach these requirements with our 3D printer.

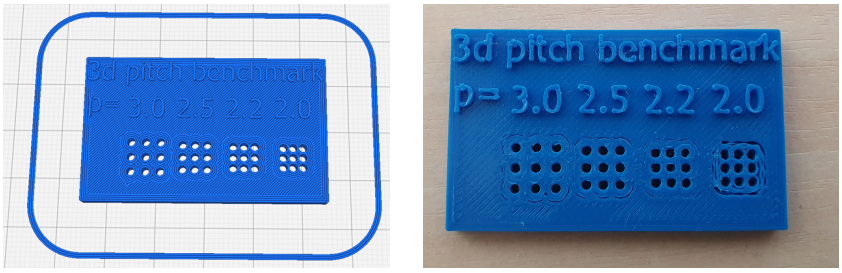
To evaluate whether we can 3D print the pin-grid, we created some benchmark prints that show us our 3D printer's precise capabilities and limitations. Firstly, we need to know whether it is possible to print 1.0 mm holes successfully. Therefore, a design was made with various hole sizes, to observe what sizes are achievable, which is shown in Figure 4.9a. These results are shown in Figure 4.9b. From this benchmark, we learned



(a) 3d printed design

(b) 3d printed result.

Figure 4.9: The 3D-print benchmark to determine the minimum hole size that our 3D printer can reasonably print. On the left, the 3D design is shown, on the right the 3D printed result.



(a) 3D printed design

(b) 3d printed result.

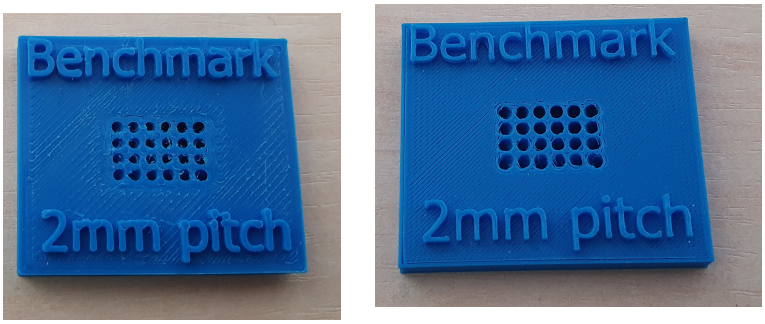
Figure 4.10: The 3D-print to determine the minimum hole spacing that our 3d printer can reasonably print.

that the smallest successfully printed holes are 0.7 mm in diameter since smaller holes were deformed significantly. So 3D printing holes with this size seems possible since the minimum hole size is more than sufficient for our requirements.

It must be stated that these tests are performed using PLA material, due to its excellent 3D printing characteristics, making precise and consistent 3D prints possible. Further, the reason that PLA was not used for printing the actuators is the low glass transition temperature of 55 celsius. This low glass transition temperature is no problem for the pin grid's fingers-end since it is not exposed to temperatures above the user's body temperature. Here we are more concerned with the print quality of the material, where PLA excels.

Next, the smallest printable grid spacing for our 3D printer was tested, using the hole size of 1.0 mm. The benchmark is created similarly to the hole printing benchmark, with smaller and smaller grid intervals. The design and the results are shown in Figure 4.10. From these images, it can be seen that a grid of 2.2 mm is just feasible, and an interval of 2.0 mm is pushing it too far. Nonetheless, we wanted to reach the better value of 2.0 mm.

To reach the desired grid spacing of 2.0 mm, modifications to the 3D printer were made. To avoid clogging the holes with stringing, filament retraction was increased from 2.5 mm to 5.0 mm, for more precision the standard 0.4 mm nozzle was replaced for a 0.2 mm nozzle, and the printing speed was decreased from 50 mm s^{-1} to 25 mm s^{-1}.



(a) Benchmark for angled 3D printed holes

(b) Benchmark for angled 3D printed holes with improved 3D printer settings.

Figure 4.11: The 3D-print to verify whether the shell of the pin-grid array can be printed successfully.

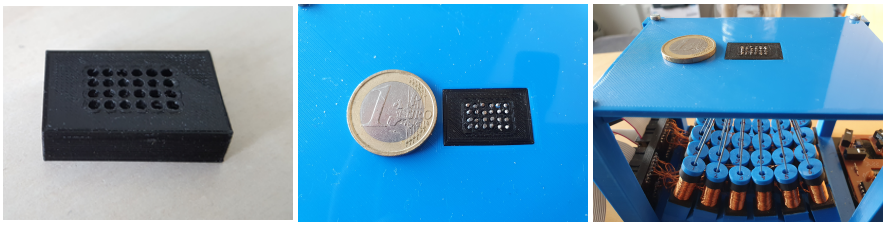
These modifications were tested on a new test print, with only holes at 2.0 mm pitch. These holes are designed with the precise angles as the holes will be in the final design. The first test that was performed showed some over extrusion (Figure 4.11a). After adjusting the feed rate down by 10 %, the results were deemed adequate. The result of this test run is shown in Figure 4.11b. It is not visible from the images, but the holes are all angled inwards towards the fingertip since the pushrods extend from the actuators placed below the finger piece. From this result, it was decided that the guiding holes for the pushrods can be 3d printed, for great control, consistency and accuracy of the final result.

4.2.2. RESULTS

Concluding the achieved resolution, we can now achieve a pin-grid resolution of 2.0 mm. This is achieved using highly consistent, and precise 3D printed guiding pieces, that guide the actuation force to the precise locations on the fingertip. Testing showed that it is possible to use 3D printed holes for the finger piece, and with some modifications on the printing process, we could reach the precision to consistently print with a pin-grid resolution of 2.0 mm. The design is modular, such that we can replace the pin-grid at the finger end, create different resolutions or a finger rest shape if necessary. The pushrods are self-locking to the magnets, realizing quick and easy disassembly or replacement of actuators or pushrods. The final finger piece is shown in Figure 4.12.

4.3. DYNAMICS

As identified in Section 2.3, the dynamics of the HTD is one of the cornerstones of proper texture interfacing to the fingertip, since it transforms the static texture display into a dynamic texture environment. The development of the dynamics poses several design challenges that need addressing. Firstly, the motion of the device must be read at a high frequency, to keep latency low and response time high. Secondly, this detection system's resolution must be high, preferably half the pin-grid resolution or less, such that the user does not notice the resolution of the motion system. Thirdly, the whole HTD must be freely movable in two dimensions over the surface (two DOF). It must do so without



(a) Final print of the replacable block for setting the pitch

(b) Block fitted to the actuator system, viewed from above of the finger-pad

(c) Side view of the finger-pad, denote the pushrods pointing inwards

Figure 4.12: The final 3D-printed pin-grid system viewed from various angles (euro coin for scale).

obstruction or significant friction of the free movement.

4.3.1. SENSOR SELECTION

To realize the movement detection system, the first thing to consider is what sensor will be used to monitor the movement. As detection systems, we have looked at numerous systems, like a direct-contact system with linear resistances, Time-of-flight (TOF) sensors, ultrasonic distance sensors and even lidar sensors. Nevertheless, none of these sensors is adequate for our specific purpose. A direct-contact system limits the free movement, and will be a bulky solution is a surface texture area is used. An ultrasonic sensor is not fast enough and is not feasible in more than one dimension, due to interference. A TOF sensor does not have the wide range we require and is cumbersome to use in a 2-dimensional measurement setup. Lastly, a lidar setup is very bulky and expensive. However, an essential aspect of this measurement is that absolute accuracy is not of great importance, but the accuracy of relative movement is. With this, we mean that we do not need to know precisely where the finger is at any given moment. We only need to know where it is moving towards precisely. The movement is essential; the location is not. And then it occurred that what we want from the movement detection is similar to what a computer mouse does. A computer mouse also keeps track of 2-dimensional movement over the surface.

A computer mouse is explicitly designed to detect relative movement over a two-dimensional surface, with high precision and low latency. Instead of reinventing the wheel by designing our own sensor system, we will use the existing computer mouse sensor for our dynamic movement detection system. For this purpose, we will use a PS/2 mouse since this protocol is well implementable on a microcontroller, and we do not have to implement the full USB protocol stack. This does, however, mean that we are limited in the update rate since PS/2 supports a maximum update rate of 200 Hz.

The mouse that was used for our HTD is a Logitech SBF96 mouse[2]. Further, according to the manufacturer, this mouse has a resolution of 400 Dots per Inch (DPI), meaning that the resolution of the detectable movement is 0.064 mm, which is more than adequate for our application. Lastly, the update rate of this mouse is limited, as per the PS/2 specification at 200 Hz. Lastly, there is no limitation in freedom of movement, since only relative measurements are considered.

[2]The mouse is rebranded with the NEC logo; the model name and the PCB are Logitech's

Figure 4.13: HTD prototype with a computer mouse built into the base, used as motion detection system for the dynamics

To verify the idea of using a mouse as a motion detection sensor, another prototype HTD was built, with a 3D printed compartment where the complete mouse fits in. In later designs, the mouse will be broken down, so only the necessary inner parts of the mouse are used. This saves weight and makes the system less bulky. This prototype is shown in Figure 4.13.

This prototype revealed that a mouse works well for our application. It feels responsive, and the resolution of detectable motion should be more than enough, such that is not noticeable. Further, one of the limitations of this detection system that must be noticed is that rotations of the fingertip are not detected; the only movement along the x and y directions is, as the computer mouse is not designed to include rotation in its regular movement set. We decided that this is something to consider, but it was decided that it is not a big issue for our device's functioning.

Measurements of the end-to-end latency learned that the latency of this mouse was approximately 17 ms. The complete end-to-end latency of the final device, including a measurement setup is shown in Chapter 6. With this 17 ms latency in the motion detection system, the end-to-end delay is guaranteed to be substantially longer than our aim of 10 ms. Therefore, it was decided that a more modern USB mouse must be implemented, with high polling rate and lower latency. The specific model of mouse chosen for the HTD is the Logitech G502 HERO. This modern USB mouse also improved the polling rate to 1000 Hz, and the DPI to 1600.

4.3.2. RESULTS

To conclude, the dynamics are realized using a computer mouse as a motion sensor. It gives two DOF and has a resolution of 0.016 mm in both the x-and the y-direction, as per the specification of the mouse. The movement is not limited in any way by this detection system. However, it must be noted that rotations are not detected by this system, as a result of the use of a computer mouse as motion detection system. An additional rotation sensor needs to be added to the motion detection system to include rotational movement. We determined that this was not essential for our purpose and did not add

this feature; hence our device can only be used with lateral movement. Further, the update rate limited at 1000 Hz, which is limited by this specific type of mouse. Including the complete USB protocol stack that is handled in the microcontroller, the total latency between the movement of the device and getting the movement being received by the microcontroller was lowered to 7 ms.

4.4. COMPATIBILITY

Compatibility of the HTD determines the device's usability, so direct compatibility with existing data is desired. With the compatibility of the HTD we mean as explained in Section 2.4 that the device must be able to load texture is a data format such that existing textures can be used. An often seen format for texture data, which is specifically prevalent in the gaming industry is the use of heightmaps. The advantage a heightmap is that the texture is not stored with polygons, but is stored as hights at any location on the surface, saving compute power and memory space.

4.4.1. HEIGHTMAP

A heightmap is a data storage format widely used to digitally store elevation values of landscapes, surfaces, and textures. Heightmaps are widely used in the gaming industry, terrain rendering programs and landscape viewing programs. A heightmap technically is an image format, where every pixel of the image stores an elevation value of the surface. Often, grayscale values are used for the elevation values; black is the minimum elevation level or the "floor" level, and white is the highest or "top". Then, the relative pixel brightness specifies the displacement height from the floor level for that position.

When a grayscale image is used, each point's resolution in the heightmap is stored one byte, therefore has 256 possible values. When more resolution is needed, the heightmap can also use the standard RGB 8-bit colour space. This means three bytes per pixel, or 2^{24} possible values per position.

Since a heightmap is fundamentally an image, it can be created manually, using any simple paint program. There exist also specialized terrain editing programs to generate heightmaps. Further, it is possible to generate heightmaps by using polygon meshes and transforming them into a heightmap. Real-world terrain heightmaps are often automatically generated from measurement systems, for example using radar systems. An example heightmap is shown in Figure 4.14, where a heightmap of the landing site of the Apollo 11 is pictured.

Another advantage of using an imaging format for storing the data has the advantage that heightmaps are easily visualized by any image viewing program. To use heightmap images for our HTD, we must be able to read these images and import them in the microcontroller. Therefore, we created a python script that transforms the .PNG heightmap image format to an array with floating-point values between zero and one, corresponding to the black-level of each of the pixels in the image. This array is imported in the microcontroller, where it has fast read access for driving the pin grid.

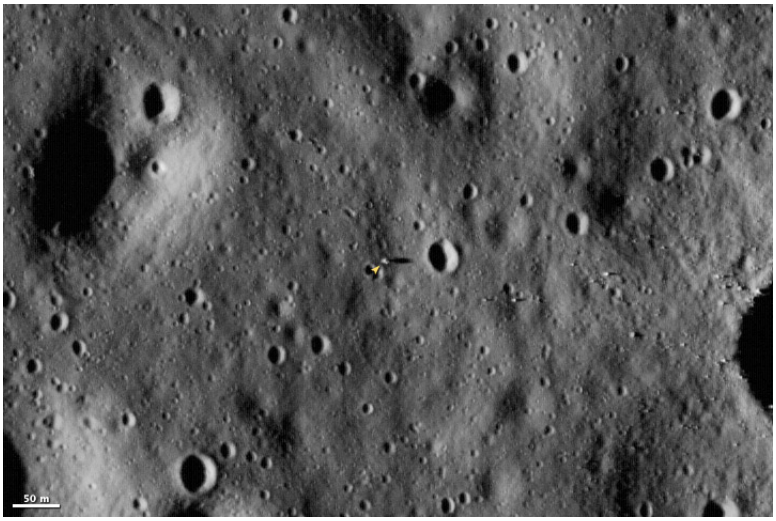


Figure 4.14: A heightmap of the NASA Earth Observatory, where the landing site of the Apollo 11 on the moon is shown, captured by the Lunar Reconnaissance Orbiter in 2009.

4.4.2. MICROCONTROLLER

It is also necessary to determine what controller will be used to drive the HTD. We chose to use a microcontroller over solutions like FPGA or hardware due to the flexibility, speed and wide availability of ready to use development boards. We made a few selection criteria for choosing a fitting microcontroller board. We need 24 PWM outputs, a few spare GPIO-pins for the motion sensor, and a communication interface for sending and receiving data from the computer. Further, enough memory is needed to store the heightmap and to store data logging data. It is uncertain how much memory is needed exactly, but we estimated that one megabyte of flash and 100 KiB of RAM should be ample. After some searching around, we found the STM32 Nucleo-F767ZI board. It fits the requirements with ease; It has 2 MiB of flash, 512 kB of ram, and up to 30 PWM outputs and more than 100 IO pins. Since it is also competitively priced and available from many web stores, this was the microcontroller board we chose for this project.

4.4.3. VISUALIZATION

We wanted to visually show what the working of the HTD when in use. It is desirable if even if someone who has never used the device can understand what it does. We wanted to use the heightmap as a visual image, and show the user's finger movement is relative to the image. To do so, we created a data logging function, where the microcontroller stores the movement data from the mouse sensor, together with timestamps for every change. In practice, this means that the data logging logs at 200 Hz, since this is the frequency that the data from the motion detection is received. This stored data can afterwards be sent to the computer, using the COM port. On the PC side, we created Windows software that reads these timestamp and location data. The program can combine this data with the heightmap, and visualize the movement over it by a transparent box, that represents the

Figure 4.15: A hand-made heightmap image, used for preliminary tests of the HTD.

HTD moving over the surface. A screenshot of this visualization is shown in Figure 4.15. It is also possible to export this visualization to an image sequence. This image sequence can be converted to a video using video montage software. This makes for a video that showcases the use of the HTD in an intuitive and straightforward fashion.

This program was tested and was deemed not intuitive enough to visualize what is going on. Therefore, we changed the visualization method, now instead of a static image with a moving finger over it, the other way around. Now, the finger is statically in the middle, and the surface is moving under it. From our perspective, this gives a clearer view of what is going on. Further, in the video editing software environment, we place the actual video of the person using it next to the visualizing graphic. In the video, we can simultaneously see the device being moved by the operator and graphic visualization of the device moving over the heightmap. In our opinion, this gives a more logical representation of what is happening.

4.4.4. MODIFYING HEIGHT MAPS FOR THE HTD

We chose to make use of heightmaps to serve as the input data for the HTD. This is a logical desicion as heightmaps are readilly accessible from different industries. However, the HTD has one problem with regard to representing the height map information. Heightmaps express a displacement with respect to a reference plane. The HTD doesn't produce displacement, but instead it applies force, pushing into the fingertip. The displacement that results is a combination of the pushing force and how the fingertip deformes because of that. Another complication is that not only the shape of the finger tip, but also the amount of downward force the operator applies with its fingertip plays a factor. Considering these things, the translation between displacement and force doesn't have to be linear.

While performing experiments with the HTD, we discovered that the difference between no and some force feedback is significantly more noticeable than the difference between some and a lot of force feedback. As a consequence, textures that have more parts that have zero feedback are more easily distinguishable. The problem arises when the texture has negative spaces like holes or crevices. The HTD cannot convey negative values, so in order to be able to show a negative value, the baseline is lifted up. This effect is illustrated in Figure 4.16. In this Figure, both A and B have a peak of 0.5 height, with

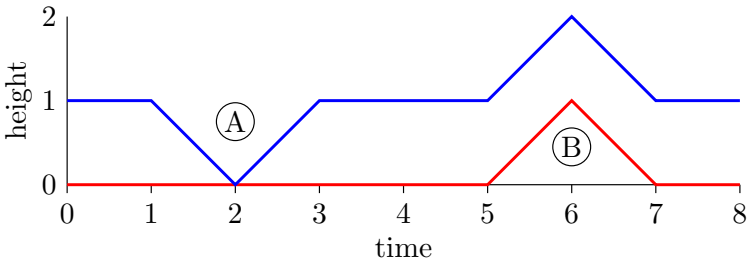


Figure 4.16: The red line shows a heightmap with one peak. The blue line shows a heightmap with both a peak and a hole. To portray the hole at (A) the baseline is increased so that the texture data never falls below zero. At (B) both height maps have a peak. It turns out that the red peak is much more noticeable than the blue peak because the difference between no and some force feedback is significantly more distinguishable than the difference between some and a lot of force feedback.

the only difference that A also has a hole of 0.5 deep. Because the HTD cannot produce negative values, to portray the hole, the baseline is lifted. However, this increase is at a cost, as the peak in B is significantly more distinguishable than the peak in A even though they have the same height.

To improve the ability of the HTD to convey heightmap data, there are two main options. The first option is to include control logic and distance sensors to modulate the force output. This theoretically allows the device to produce a displacement output effectively. The second option is to perform additional processing to the heightmap data to make it more suitable for the HTD. The former option is an interesting possibility, but it is out of scope for this work. The second option will be further pursued in this Section.

As mentioned before, the difference between zero and some force feedback is significantly more noticeable than the difference between some and a lot of force feedback. Therefore it would be a good idea to maximize the instances where the texture data is conveyed through the difference between zero and some feedback. We propose a method to process heightmap data into a format that does exactly that. This method is based on the assumption that for this HTD low-frequency texture information is insignificant compared to high-frequency data. The concept behind the method is to modify the low-frequency data so that instances where high-frequency information is portrayed in the effective region. Here the effective region is the zero to some force feedback region.

Implementation

The implementation is split into four parts. The entire process is illustrated in Figure 4.17. **First step:** Make a copy of the original heightmap that is blurred using a gaussian blur. The resulting image represents the low-frequency components of the original heightmap.

Second step: Subtract the gaussian image from the original. The resulting image has its low-frequency components removed, so only high-frequency components remain. This image has negative values in it. To illustrate negative values, they are shown in blue, while positive values are shown in red.

Third step: The obtained image in step two has negative values, but those cannot be

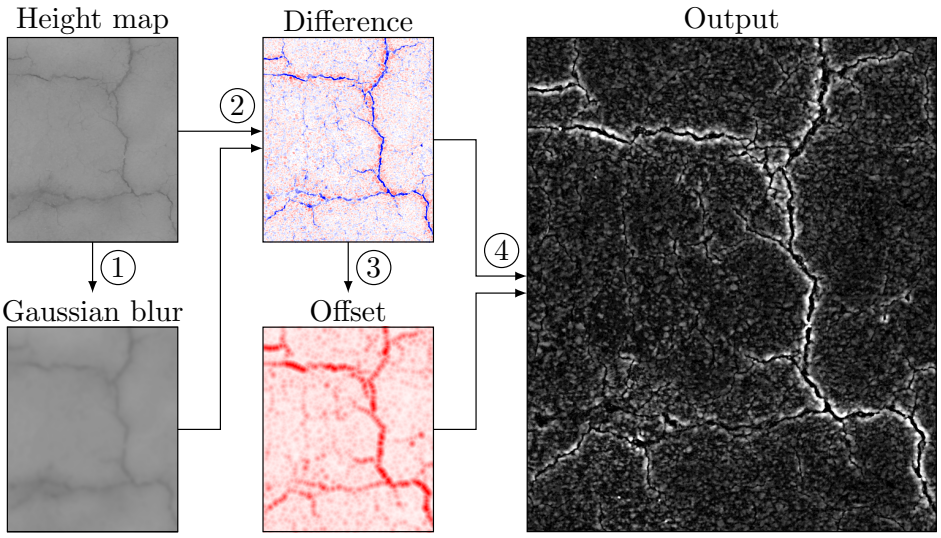


Figure 4.17: Example of how a heightmap is modified for the HTD. For the heightmap, gaussion blur, and output images, the amount of white in each pixel indicates height. For the difference and offset images, the amount of blue indicates the amount of negative value, and likewise the amount of red indicates the amount of positive value.

portrayed by the HTD. To still be able to show holes and crevices, the areas around the negative values should be raised. Here we again make use of the assumption that low frequencies are not important. Where in step 1 and 2, we used that assumption to remove low frequencies, here we add some low frequencies back in to enable the negative values to be portrayed. An algorithm was designed to calculate an image with offsets based on the negative values in the difference image. This algorithm will be described later in this Section.

Fourth step: The difference image and the offset image are added together because of how the offset image was calculated, previously negative values end up at approximately zero. Because of this and the previous step of removing low frequencies, the amount of very low or even zero height in the new heightmap is increased significantly. In Figure 4.17, one can see that the original heightmap is mostly grey with some darker crevices. The output in sharp contrast to the original is significantly darker, which indicates force feedback being closer to zero. In the areas where there is a crevice, a white aura is added with a sharp decline where the actual crevice sits. Because the transition close to the crevice is so sharp, the softer transition leading up to the needed height is experienced in a less significant way.

Offset algorithm

We developed an algorithm to calculate the offset image deployed in the third step in Figure 4.17. The idea is to create a cone-shaped pedestal for every pixel that holds a negative value. The range of the cone is fixed, and the slope is directly proportional to the

(a) Difference (b) Offset (c) Output

Figure 4.18: An example of how the offset algorithm works. Blue and red color indicate negative and positive pixel values respectively. The output is the sum of the difference and offset images.

negative value for that pixel. A constant slope was tried before a constant range; however, this method has problems, because when there are small negative values, it results in a small positive cone. This small cone has a lot of high-frequency components and therefore introduces a noticeable bump that was originally not present. A cone with constant range works out better because of how noticeable the cone scales, proportionally with how negative the value is. The cone can almost not be noticed for small negative values, but the crevice itself is also not pronounced. In the case of a large negative value, the cone is also noticeable, but the transition from high to low is so strong, that it is much more significant than the slope of the cone. An argument can be made that for sufficiently large negative values, the range of the cone should also be increased, but this is future work. The cones of all negative pixels are calculated separately. When the cones are combined together, the overlap is resolved by picking the highest value. An example of how the algorithm works is shown in Figure 4.18. In this Figure, (a) shows a different image. The offset that is calculated based on the difference image is shown in (b). Finally, the output, obtained as the summation of both the difference and offset image is shown in (c). Especially in the output image, it can be seen that the size of the aura for the two blocks is equally large, but the intensity is larger for the block that is more blue. Here more blue means that the hole is deeper. One can also see how overlap is resolved where consistently the highest value is picked. The output shows the intended effect, where if a participant would move towards one of the blocks, the height would first gradually increase and then sharply decline. The gradual increase is not easily noticed, but the sharp decline is very clear.

4.4.5. RESULTS

To conclude, the compatibility of the HTD working as intended, and we have achieved the following results. Heightmaps are used as the main datastructure to load into the device since they are widely used standard for storing texture, landscape and other surface height data. This also means that we can import or create our own testing textures in pretty much any image creation or editing environment. The black and white level corresponds to 0 and 100% output force on the actuators respectively. It is possible to define

how many pixels are between the 2 mm pin grid of the pushrods for scaling the image to the fingertip. This adjustment capability is also useful when experiments are performed with different grid-spacing of the pushrods. Further, during use, the device can datalog the user's movement. This movement data can be used to visualize the workings of the HTD and for demonstration purposes. To visualize this movement, we created a software program that shows the user's finger movement over the heightmap. This visualization function is, however, not in real-time and limited by the memory of the microcontroller. We had no communication bus that was fast enough to send this data real-time without constraining the microcontroller to much to properly function, hence the data is stored locally. Therefore, the data logging is limited to about 60 s of recording. This could be improved by adding a databus, which can send data to the pc at least 1.5 KiB per second with low microcontroller overhead. Further, since the heightmap is stored in the microcontroller's main memory for fast access, we are limited to texture images of 1 megapixel in size. This limitation could be solved by adding more fast accessible memory (>100 µs) to the microcontroller, either internally or externally.

Further, to get a feeling of the visualization software, viewing a screenshot of the visualization does not do this function justice. A video of the visualization function is needed to show what is going on clearly. Therefore, videos of the visualization function are posted on Youtube, linked in [13]. Here the device itself and the visualization function are shown in action. Note the stationary blue cone, above the device. This is an arrow that is used as a reference point. In this case, the heightmap consists only of one single high spot, since this is the easiest visualizable texture. The blue cone was beforehand precisely moved to the location of the black dot on the "surface". Lastly, since we want the pushrods to be visible while moving, we cannot use our finger on the fingerpad. However, this means that the pushrods have no resistance in their movement and move up too far and not retract. Therefore, we placed transparent tape over the fingerpad, limiting the movement of the pushrods. Nevertheless, the movement is visible, making this a simple and comprehensible demonstration in our opinion.

4.5. THE FINAL DESIGN

Now all the separate parts of the HTD have been explained, we will focus on the combination of all these parts together, that form the texture interface as a whole. This results in the setup shown in Figure 4.19. As visible from this image, we can see the 6 x 4 actuators array at the bottom, the pushrods pointing up to the 3D printed finger rest, and the mouse sensor at the side. We have not elaborated on the electrical driver schematic, since the specific circuit that was used is not of importance for the achieved results. The schematic is however described in the appendix, in A.

To show the separate parts of the final design, the primary design blocks are accentuated within different coloured boxes. These parts are elaborated in the previous sections of this chapter and should not need any more explanation. The actuators are accentuated by the brown coloured box in the image. The actuators are placed on a concave structure, such that all actuators point inwards to concentrate at the finger's end. These actuators are connected to pushrods (also visible in the same brown box) that point upward to the finger piece. This finger piece is accentuated by the red box in the picture. The user can lay his hand down on the top of the blue device, and rest a fingertip on

the black finger piece in the centre. This is where all the actuators come together with a pin spacing of 2.0 mm. On the side of the device, in the green box, there is the PCB of the mouse, which functions as the movement sensor. The blue part is movable, with the ribbon cables determining the freedom of the movement. These ribbon cables conduct the driver outputs to the actuators. The electronic driver is pictured in the purple box in the image. The schematic and functional principles are elaborated in appendix A. Lastly, the final part of the HTD is the microcontroller board, which is visible in the yellow box in the image. As is visible, the microcontroller is connected to the output driver board using a rainbow-ribbon cable. Also, the mouse PCB is connected to the microcontroller via a black cable, which is partially visible in the image.

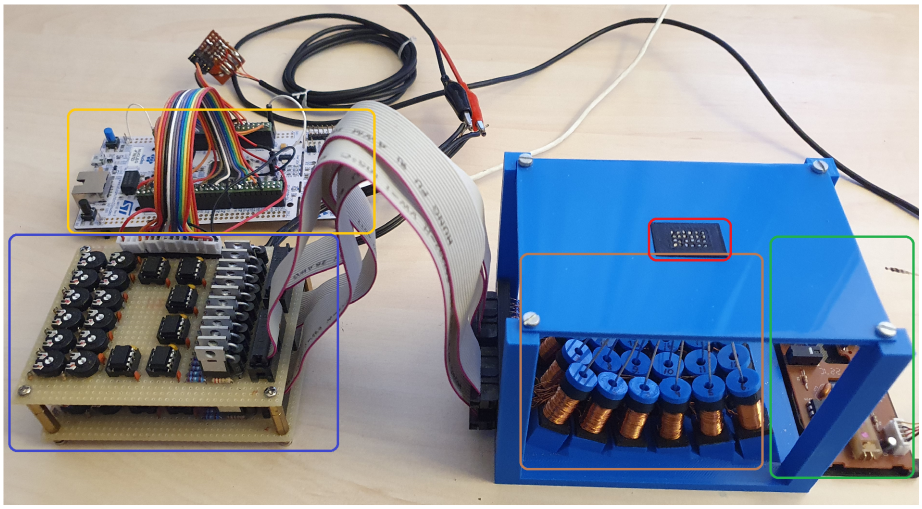


Figure 4.19: The complete setup of the haptic texture device

5

TESTING METHODOLOGY

It is important to verify how well the designed HTD performs the functions that it is designed for, and therefore want to perform two types of test. Firstly, it is necessary to verify whether the device meets the design specifications, such as the end-to-end time delay. This testing methodology is elaborated in Section 5.1. Secondly, we need to quantify how well the device performs in interfacing texture to a human fingertip. The testing methodology for this evaluation is elaborated in Section 5.2.

5.1. END-TO-END DELAY TEST

One of the important parameters of our HTD is the system's end-to-end delay. With this, we mean the delay between the device being moved, and an actuator moving in response. To test this delay, we need to move the device instantly and measure the actuator's output. Measuring the actuator movement is no problem; this was already done in Section 4.1.3. However, the movement of the HTD is a different story. We can similarly use an oscilloscope as in the actuator speed test, where the oscilloscope is triggered as soon as we pose an input to the system, and stopped when the output is given. We measure the difference in time using the oscilloscope, which gives us the end-to-end delay. The problem then is, how can we create an almost instantaneous movement on the device, and create a signal at the same time. We could use a heavy metal object, which contacts the HTD to generate the signal, and at the same time push it away to create the movement. We would need a heavy metal object, that can be used to create a hard impact on things it hits. This is shown in Figure 5.1

This image shows the solution we came up with to measure the end-to-end latency on the HTD. The hammer is used to impact the HTD, and almost instantaneously move it over the surface. Further, the side of the HTD is partially covered in aluminium foil, connected to the oscilloscope. Since the hammer is connected to +3V3, a signal is generated when the hammer touches the device. This measurement is repeated 50 times, and the average is taken to improve the accuracy of the measurements. The data obtained using this measurement setup is further elaborated in the Chapter 6.

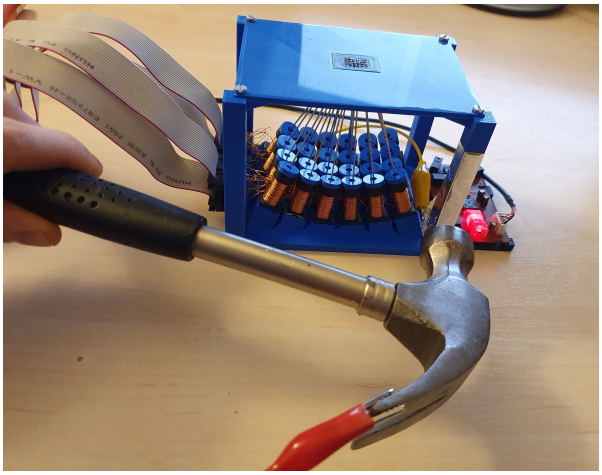


Figure 5.1: Test setup for measuring the response time when the device is suddenly moved. The hammer is used to generate a fast change in velocity on the device. At the precise time of the impact, a closed circuit is made between the hammer and the aluminium foil on the HTD. This triggers the oscilloscope to start measuring the time until the microcontroller acknowledges the movement using the motion sensor.

5.2. HAPTIC TEXTURE BENCHARK

It is important to verify how well our HTD portrays texture to the human fingertip. We need to benchmark to quantify how well the device works. However, since this is the first device to be a realistic interface for surface textures, no standardized method for characterizing the performance exists yet. We, therefore, developed our own testing method to verify how well our HTD fulfils its purpose. The benchmarking methodology is general-purpose such that similar devices, including those of competitors, can compare their performance against ours. Also, it gives us further insight into possible limitations or flaws in the performance. To verify the device's performance, we need some tests that represent challenges in the realistic operation of the HTD.

Testing goals

The benchmark to be developed should reflect real-life use. Therefore we need a clear understanding of what needs to be accomplished with the testing that will be performed. The results quantify how well participants recognize the texture that was portrayed by the HTD. For this texture recognition benchmark, it is required that the participants recognize textures since that reflects the most realistic use case. To quantify how well the participants recognize the textures that the device interfaces to the fingertip, we use objective (multiple-choice) questions. We use objective questions since they can be directly used for grading.

Getting the right textures

The textures in the benchmark can thus not be chosen arbitrarily; there are a few aspects key in selecting the right textures since we require the benchmark that uses textures that can be used in real-life use cases. Further, the distance between the pixels

in the heightmap must be less than 1 mm, in order to have enough detail in the texture to be considered as realistic. This implies that the scanned texture's size determines the required resolution and that smaller surfaces require less resolution to meet the required level of detail. Further, too much dynamic range must be avoided. Such as deep ridges or high spikes since these cannot be interfaced realistically since the pin movement is approximately 2 mm. It was designed for surface texture, not for showing mountains and valleys. With these selection aspects, it would be preferable to generate our own heightmaps from a surface since it gives us great freedom in what textures we precisely want (e.g. detail, size, material). Another advantage of making our own texture heightmaps would be that we would not be dependent on licenses when we want to share the images. Unfortunately, we do not have the required tools to create our own heightmaps from a surface. As an alternative, we obtained textures from an external source. Textures can be obtained online, from texture databanks for the film and gaming industry. One last point in the image selection process is that it is convenient to have a graduation in the final image recognition scores. The textures are strategically chosen, so there are two pairs of similarly structured patterns, which are intensionally challenging to differentiate. The intension behind this system is that it creates a gradient in the results in the case that participants can determine the rough details in the texture, but miss the finer structures that differentiate the two similarly structured textures.

Testing Process

We assumed that the participant does not have prior experience in using the HTD. Thus, for participants to get to know the device, how to hold it and what haptic feedback to expect, a warming-up program is developed. The participant gets explained how to hold the device, and how it can be used, and then the warming-up program is started. In this program, the participant can select only black and white structures, and use the HTD to feel around to verify whether they can confirm that they feel the pattern they would expect. They can select six different black and white textures, shown in appendix B. Only after the participant is confident in using the device, the real test is started. For each question, the participant is shown four images of different surfaces, e.g. stone, sand, concrete and asphalt. The participant can choose between photos of the textures because the participant should recognize the image, not recognizes the heightmap. The participant then gets 30 s to feel around and explore the surface, after which the HTD turns back off automatically. Then, the participant is asked which of the four patterns was portrayed. This same sequence is repeated for every question. An example question is shown below, in Figure 5.2 The complete question form for the texture recognition benchmark, as given to the participants is shown in appendix C.

Step	Time	Notes
Explain how the device works and how to use it Let the participant use the device while running the warming-up program	~5 min ~5 min	
Program the HTD with the next question Let the participant feel the texture Let the participant answer the question	30 sec 30 sec 30 sec	Repeat this as many times as there are questions
Total	~28 min	

Table 5.1: The sequence of steps in the haptic interface benchmark.

Example question
Select which of the following were portrayed.

Figure 5.2: Example of possible textures set for a question. Notice that the top two and bottom two images are matched (top images both have random bumps and valleys, bottom images both have regular patterns by the joints between the tiles).

6

RESULTS

To characterize the performance of the developed HTD, we performed two different types of tests, of which the methodology is described in Section 5. Firstly measurements, whether the device meets the specifications that it was developed for, elaborated in Section 6.1. Secondly, benchmarking quantifies how well the device performs in portraying texture to a human fingertip, which is elaborated in Section 6.2.

6.1. END-TO-END DELAY TEST RESULTS

To measure the end-to-end delay of the developed HTD, we performed measurements according to the methodology elaborated in Section 5.1. This method measures the motion detection system's latency, combined with the other measured parameters to get the end-to-end delay. The average end-to-end latency of the motion detection system was measured to be 7 ms. Further, the actuators have a latency of 0.6 ms, actuator driver 50 µs and microcontroller heightmap processing 6 µs.

The latency of every element in the chain is now known, meaning that the end-to-end latency can also be determined. This way, it was determined that the total end-to-end latency is 7.7 ms. Further, to visualize the relative latency of every element, a pie chart is shown in Figure 6.1. From this chart, we can see where the most significant improvements in latency can be made. As visible from the figure, the latency of the motion detection is the most significant latency source, which is a point for improvement. Reducing the end-to-end delay can still be done by using a different computer mouse that is specified even lower latency and high polling rate, for example, a gaming mouse such as the Razer Avalon [6], which claim to have a maximum latency of 0.25 ms. This mouse can still be used on this microcontroller. Therefore, there is no need to drastically change the hardware to make this work and to potentially lower the end-to-end latency even below 1 ms.

The end-to-end latency of the developed HTD, we aimed to be below 1 ms. It is clear that this number is not reached as of yet; however, we want to emphasize that every element in the chain except for the motion detection system is well below this target. All

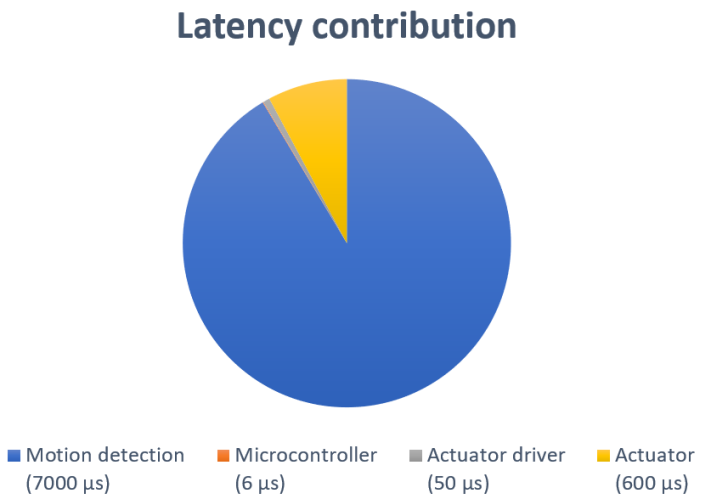


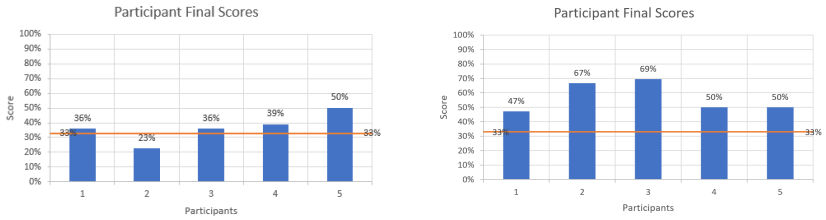
Figure 6.1: The latencies of the separate parts of the HTD. As is visible from the figure, the motion detection system dominates in the total latency. This is a large point of improvement for this device, and can be resolved by changing the mouse to one that is specified for having low latency and high polling rate, such as a gaming mouse.

the elements that we developed and designed meet the requirements, and when a mouse was to be used with 0.25 ms latency, there would be time to spare. This is a positive thing since we could with only minor hardware changes vastly improve the latency of the HTD. If with 0.25 ms latency in the motion detection system, the total end-to-end delay would only be about 0.9 ms.

6.2. HAPTIC TEXTURE BENCHARK RESULTS

Texture recognition benchmarks were performed by participants, to quantify the texture portraying performance of the HTD, according to the developed methodology given in Section 5. The benchmarks were performed with the heightmap of the textures directly stored on the HTD since the heightmap modification algorithm of Section 6.2a was not developed yet. The results of these benchmarks in percentages are shown in Figure 6.2a. The percentage shows the scores of the participants, with respect to the maximum possible score. In this figure, we also added an orange line which shows the average score that would be the result when the questions were filled in randomly. To calculate the expected percentage of randomly filled in questions, we consulted equation 6.1, which an expected percentage of 33%. We desire our average scores to be definitely above this line; otherwise, this would indicate that something is wrong with the workings of the device, as the participants would essentially be guessing.

$$\mathbf{E}[x]_{\%} = \frac{\mathbf{E}[x]}{P_{max}} \text{ and } \mathbf{E}[x] = \sum_{i=1}^{k} x_i p_i \tag{6.1}$$



(a) Benchmark results for the developed HTD, without heightmap modification algorithm.

(b) Benchmark results for the developed HTD, with heightmap modification algorithm

Figure 6.2: Performance scores on the Haptic Test for individual participants. The orange line shows the baseline.

From the benchmark results in Figure 6.2a, it became clear that the HTD did not work as well as we had hoped. We see in the scores that most of the participants score close to randomly filled in answers. These results indicate that the HTD does not correctly portray texture to the participants. We did know that the heightmap to force translation is not 1:1, as is explained in Section 4.4.4, but the expectation was that this would not have such a large effect on the performance. Therefore we developed and implemented the heightmap modification algorithm of Section 6.2a. With the new algorithm in place, the benchmark scores increased significantly, and all of the scores are safely above the orange line. These results are shown in Figure 6.2b. Here we can see that the scores have increased, such that all they are all safely above the orange line, and the average score increased from 37% to 57%[1].

From Figure 6.2b, it might seem especially for participant 1 that the score is low, close to randomly filled in answers. Table 6.1 however tells us a different story. This table shows for every individual question, what answer was given and whether the answer was correct or not. The colours in the image work as follows; No colour: Incorrect, Yellow: Correct Pair Wrong Answer, Green: Correct Answer. From this table, we can see that even participant 1 has felt the patterns clearly, only was often unable to distinguish the fine detail between the closely matched image pairs.

Further, Table 6.1 also gives us insight into what textures were recognized by many participants and what textures turned out to be more challenging to recognize. This table gives insight into what are possible strong and weak points into the design. For example, we can see that all participants had question 1, image 4 correctly recognized. Using this information, one could conclude that these types of images, with rough transitions from high to low, are well-recognizable. These conclusions are what this table is useful for, but we want to wait until more testing is done to state these types of conclusions confidently.

[1] A note is in its place here since yet only three participants performed this benchmark since software change. After this software change, only limited time was left for new verification tests with participants. However, the test has been done five times as of yet, and the results are at least promising.

Question	Image	Partic. 1	Partic. 2	Partic. 3	Partic. 4	Partic. 5
1	1				1	4
1	2	1			4	
1	3		3	3		
1	4	4	4	4		3
2	1	2	1			1
2	2	1				
2	3			4	2	
2	4		4	1	4	4
3	1		1			
3	2	1		2		4
3	3		2		2, 3	
3	4	4		2		1
4	1		4		2	
4	2	2	2	2	3	2
4	3					
4	4	1		4		3
5	1			1	2	
5	2	1	2	2		2
5	3	2			3	2
5	4		1			
6	1	3		4	1	
6	2	2	3	2		
6	3		3			3
6	4				1	3
Scores		**17**	**24**	**25**	**18**	**18**

Table 6.1: Answers of the individual participants and the correctness.
No color: Incorrect, Yellow: Correct Pair Wrong Answer, Green: Correct Answer

7

CONCLUSION AND DISCUSSION

7.1. DISCUSSION

This thesis has been unconventional due to the complications imposed by the COVID-19 pandemic. This pandemic played an important role in many aspects of the thesis, such as contacting professors, working from home and working with limited resources. This also played a role in the fact that only a limited number of participants are used for the texture benchmark of this thesis. The COVID-19 pandemic made it challenging to gather participants for texture benchmark in a short time span.

Other than pandemic-related limitations, there is also something to say about the motion detection system's latency. Although we already have a theoretical solution for this limitation, the implementation would require the motion detection system to be changed, and the software to be modified. Since the new benchmarking with participants took much of the time in the last stage of the project, this improvement is left for a next revision Haptic Texture Device (HTD).

Lastly, we want can state that the use of a HTD does not deliver its full potential when it is not connected to an input device that scans the texture on the other side of the network. There is still no way to perform remote operations without a device on the other side. However, we decided that this is outside the scope of this project. It would nonetheless be a great addition to the HTD and increase the scope of possible applications greatly.

7.2. FUTURE WORK

The idea is that the HTD gives rise to numerous research opportunities. A few of them are highlighted in this section.

First of all, the device currently does support a demo Tactile Internet application. For example the HTD does not support real-time communication, beit to a computer or even a device connnected through the internet. The simulated texture is stored locally and loading a new texture takes significant time and requires flashing a new program to the microcontroller. For the HTD to partake in Tactile Internet applications it is essential

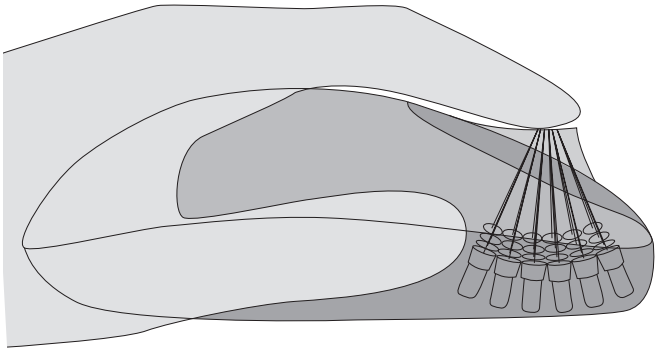


Figure 7.1: Concept of embedding the HTD into a conventional mouse. This form factor would significantly improve the produced device

that this aspect sees improvement. When communication is added to the device, the next challenge is what the device should be communicating to. The remote device that serves as the digital twin to the HTD needs to be able to grasp detailed texture information in extremely harsh latency requirements. There exist efforts into devices that can sense texture, notably SynTouch sensors, but more effort is needed. An alternative is to connect the device to a virtual physics environment where the texture data can be more easilly obtained and communicated.

The second aspect that could use work is the feedback the device produces. In the current format the force feedback is mostly a weakness as it tries to convey displacement data. Certainly force does not feel the same as actuators that can produce displacement. However, long term, force feedback is the better option. When the device is presented to more dynamic environments displacement is too limiting. For example, imagine a ball dropping on your fingertip, you would expect a force to hit your finger. One can try to mimick the effect with displacement but pulling of a realistic feeling would be extremely challenging. Using force as a starting point to realistically convey displacement is more feasible however. If one can make use of a form of feedback loop by sensing the extension of each pin, a control system can be used to mimick a displacement through modulating the force. What should also be noted is that the device allows for sensing textures on surfaces that are moving independently of the user, something that is also difficult for displacement actuators as they are generally much slower. A second opportunity with force feedback is the use of vibration. When dragging your finger over glass or over rubber, it will feel different. When dragging the finger over rubber in particular, the finger will vibrate in a characteristic way. A future endeavour would be to add such vibrations to the feedback so the texture is better experienced when dragging over it. Similar to this we expect that through vibration and careful modulation of force also other aspects of touch can be accurately recreated.

The third aspect that we would like to improve is the form factor of the device. Currently its a big heavy box, that doesnt really fit the finger tip. It would be nice to sport a different more elegant form factor that significantly increases the ease of use of the device. One direction this could go in is to make use of the general concept of a computer

mouse. In the HTD mouse parts are added to the device. Another approach is to add parts of the HTD to a mouse. An illustration of this idea is shown in Figure 7.1.

7.3. CONCLUSION

In this thesis, we addressed the research question "How can we develop a system which convincing and dynamically interfaces digitally represented textures to the fingertip?". To realize a Haptic Texture Device (HTD) we identified four primary requirements. The first requirement is to use actuators that produce force feedback with low latency < 1 ms and at least 24 mN per mm^2 of pushing force. These requirements were met using a novel actuator design based on an inductor magnet system, which achieved a latency of 0.6 ms and 50 mN per mm^2 of pushing force. The second requirement is to construct a 2d grid of individually controlled actuators, that push into the fingertip with at most a 2.3 mm pitch. The developed HTD utilizes a concave structure to increase the spacing between actuators and minimize magnetic actuator interferences. Metal pushrods, guided through a custom 3d printed grid structure yield a 2.0 mm grid. The third requirement is that when the device gets moved, the portrayed texture is updated accordingly, within 1 ms of latency. This requirement is realized by repurposing a computer mouse as a motion detection sensor. This system gives us 2-DOF where the device can be moved over a flat surface. The measured latency is 7.7 ms. However, we identified a potential solution using a low-latency specified mouse. The delay of our HTD except the motion detection system was measured to be 0.65 ms, leaving headroom for a low latency motion detection system. The fourth requirement is to support a readily available method to obtain texture data. We chose to use heightmaps, as used in the entertainment industry, as texture data. Additionally, we developed a novel algorithm that modifies the heightmaps, so they are better represented by the HTD. The final product is shown in Figure 7.2.

Further, we developed a novel benchmarking for HTDs, that quantifies how well participants can recognize haptic textures. The benchmark is general-purpose such that similar devices can compare their performance to ours.

The results of our benchmarking initially indicated that the device was not working as intended, since the participants scored 37% on average, which is close to the average random guessing score of 33%. The algorithm to modify the heightmaps was added and yielded an average score of 57%, which is a substantial improvement.

With these contributions, we present a novel HTD, the first of its kind, and a novel benchmark to quantify the performance.

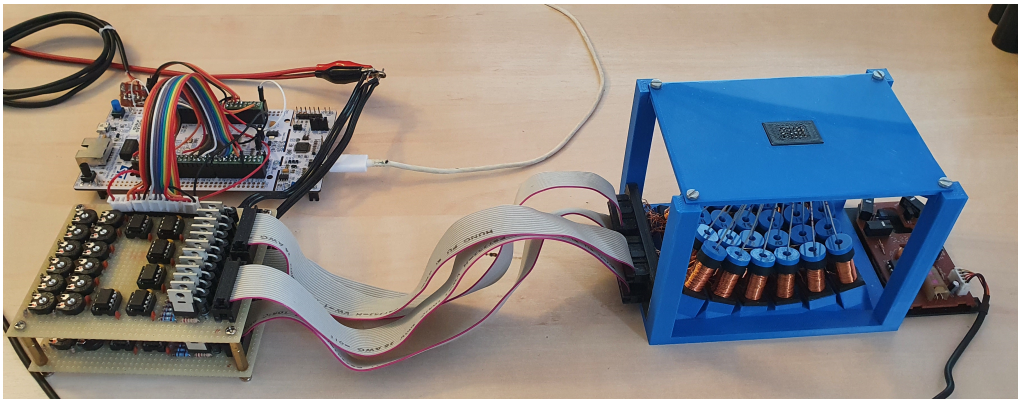
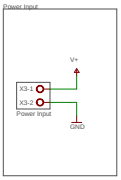


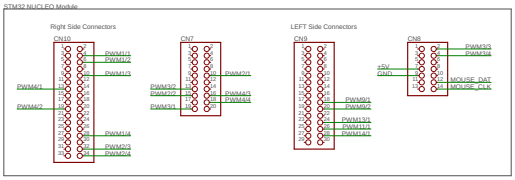
Figure 7.2: The final design of the haptic texture device.

A

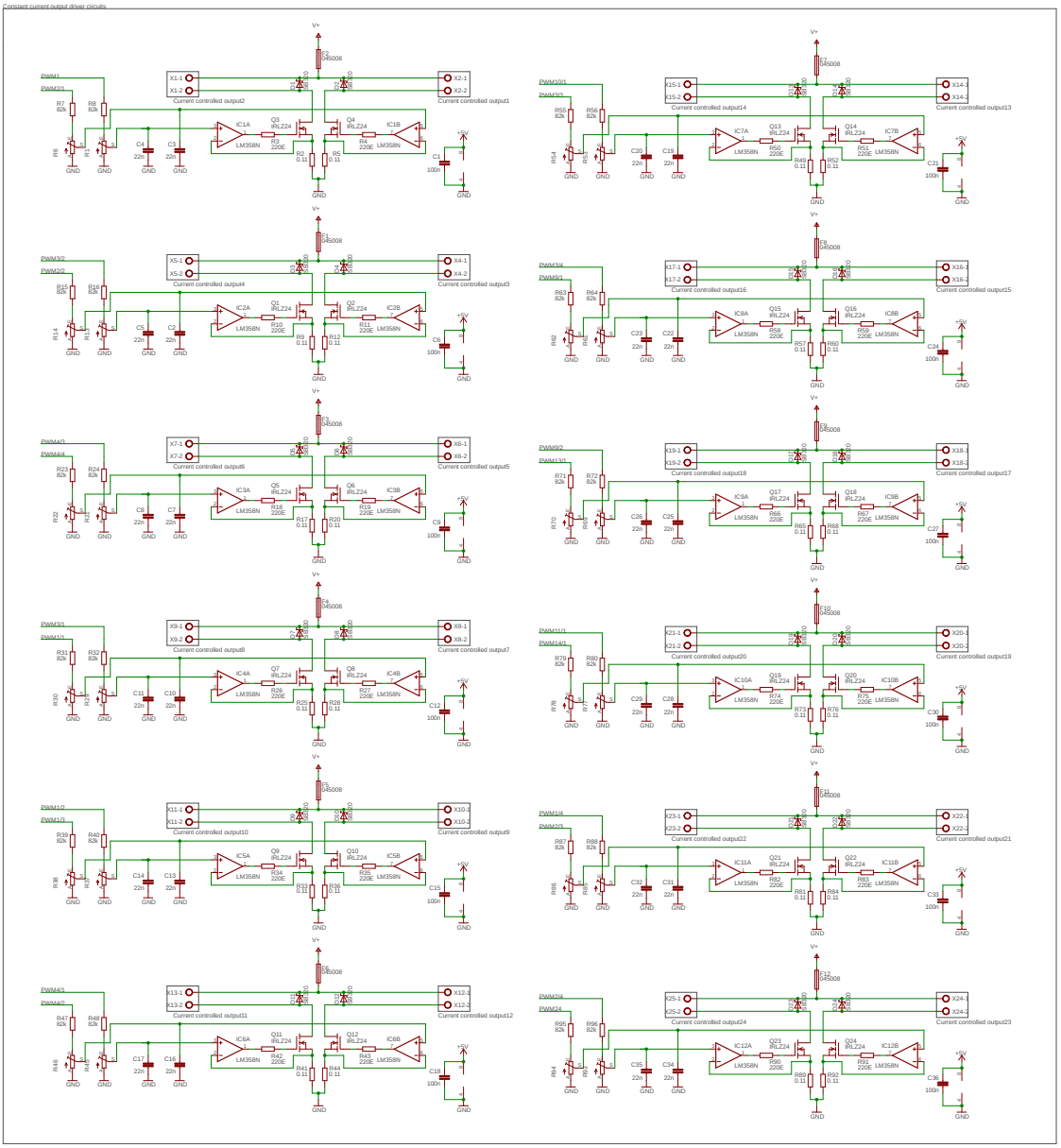
ELECTRICAL DRIVER SCHEMATIC

This appendix shows the constant-current actuator driver schematic. A current-sense resistor provides individual current feedback on every output. The voltage across it is measured, and according to $V = IR$, the current is measured and compared to the set-point, which is set using the potentiometer. This is done using the LM358 op-amp, which was chosen due to its common-mode range, including the GND, low voltage operation, and wide availability. This op-amp drives the corresponding MOSFET in its linear region, which means that the MOSFET will dissipate the power needed to create the constant current output. When the VIN is between 4 to 5 volts, this power dissipation will be minimized while the MOSFET can still provide the set current. Lastly, the Schottky diodes are placed parallel to the output. This diode is placed there to prevent inductive kickback from damaging the MOSFETs during fast turn-off. The schematic is shown in the figure below.





Constant current output driver circuits



B

HAPTIC WARMUP PATTERN SELECTION

The figure below shows the different structures that are used as a warmup for the main texture test. These shapes are discrete by design, as these were found to be the easiest to recognize.

Warming-up Patterns:

Index	**Pattern**
1	
2	
3	
4	
5	
6	

C

QUESTION FORM SELECTION

This appendix shows the complete question form for the texture recognition test, as it was given to the participants. The heightmaps will however not be shared due to licensing; however, they are free to download [22] for personal or commercial use [23] in medium resolution (1024x1024), or higher resolution if required, be it behind a paywall. The full question-form that the participants get is shown below, and more information about these questions can be found in chapter 5 of this thesis.

Haptic texture test:

The following evaluation will involve the recognition of textures derived from various surfaces. In this test, a random texture is "displayed" on the haptic texture device. It is then your task to examine the texture and try to recognize which of the surfaces is displayed. You must select one of the four given textures that is thinks was displayed.

At the start of every question, the haptic texture device randomly selects one of the textures to display. Then, you get a half minute to find what texture is being displayed. After half a minute, the haptic texture interface turns off automatically. If this has happened, you can select which one of the four given textures you think was on display.

Every question has a group of four textures consisting of two pairs of textures. Each pair consists of two surfaces that are selected to have moderately similar textures. The two pairs are less similar to each other. This is done in order to create a graduation in the given answers. When an answer is given that is not the right texture but is of the right pair, parcial points are given due to it being almost right.

A correctly recognized texture yields three points and the right pair, but the wrong answer yields one point. If any texture of the wrong pair was answered, zero points are given. Therefore, a higher number of total points means that more of the textures were correctly recognized.

To be completed by the participant:

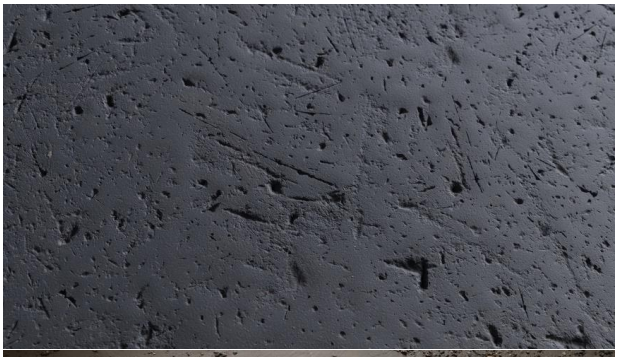
Name: _____

Date: _____

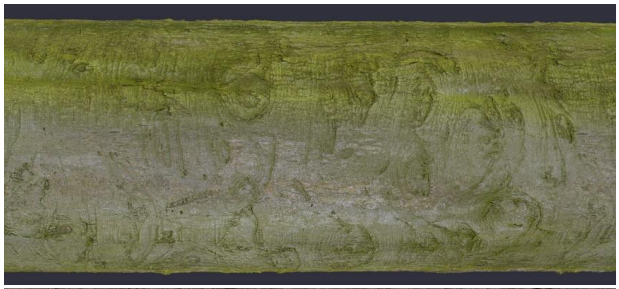
Q1. Select which of the following were displayed.

- 
- 
- 
- 

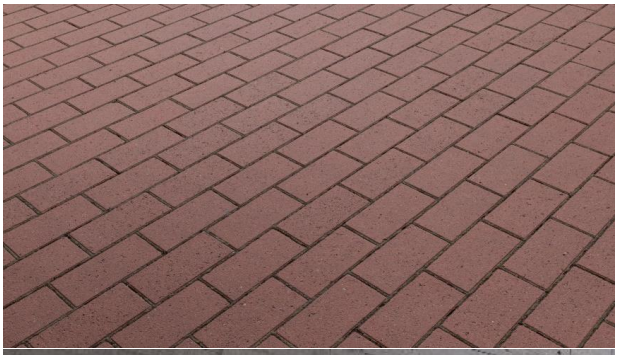
Q2. Select which of the following were displayed.

- 
- 
- 
- 

Q3. Select which of the following were displayed.

- 
- 
- 
- 

Q4. Select which of the following were displayed.

- 
- 
- 
- 

Q5. Select which of the following were displayed.

- 
- 
- 
- 

Q6. Select which of the following were displayed.

- 
- 
- 
- 

END of the test

Score table

Question	Given answer	Correct Answer	Points
1			
2			
3			
4			
5			
6			

Total score:

D

HARDWARE SPECIFICATIONS

This section outlines the information needed to recreate the device presented in this work.

D.1. ACTUATOR DESIGN

- 3D designs can be found in the git repository https://github.com/jlemmers/Thesis.
- The actuator is printed in ABS plastic because the heat resistance of PLA is inadequate.
- The rest of the device is printed in PLA plastic due to superior printing quality using PLA.
- The inductor is handmade for this version. A copper wire of 0.3 mm diameter was used.
- Each actuator coil was made by winding four layers of tightly wound wire that fills up the recessed area in the printed plastic.
- The upper part of the actuator houses a magnet (N52 neodymium) with a 5 x 10 mm cylindrical shape.
- The magnet is trapped inside the cylindrical compartment in the actuator by adding a 3D printed cap that is glued on top of the actuator. One should be careful not to get glue in the shaft, which will restrict the magnet's movement and ruin the actuator.
- Pushrods are made from 0.9 mm diameter iron wire, approximately 70 mm long.
- Depending on the proximity to the centre of the display the rods of different length.

- For this device, a rotary grinding tool was used to finetune the length of the individual rods.
- The actuator driver board requires 3 to 4 V to operate. The current consumption is about 10 A, with spikes to 20 A.

D.2. MISCELLANIOUS

- The base for the actuators consists of a separate base and top that are screwed together.
- the top part consists of two pieces where the lower one serves as a supporting structure. This structure is glued to the top part.
- The piece with the holes to guide the pushrods is printed separately and is only held by gravity on the top part of the base.

Acronyms

ABS Acrylonitrile Butadiene Styrene. 24

AC Alternating Current. 26

CC Constant Current. 26

DC Direct Current. 6, 26

DOF Degrees of Freedom. 10, 11, 30, 32, 51

DPI Dots per Inch. 31, 32

ERM Eccentric Rotating Mass. 6

FFF Fused Filament Fabrication. 22, 23

HTD Haptic Texture Device. v, 3, 4, 5, 7, 9, 10, 11, 13, 16, 17, 19, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 49, 50, 51

IC Integrated Circuit. 26

PCB Printed Circuit Board. 7, 20, 21, 22, 31, 40

PLA Polylactic Acid. 23, 24, 29

PWM Pulse Width Modulation. 26, 34

SMA Shape-memory Alloy. 7

TOF Time-of-flight. 31

TPD Two Point Discrimination. 9, 27

BIBLIOGRAPHY

[1] Global Gizmos Store on Amazon. *Global Gizmos Pin-art sculpture 18cm x 13cm*. https://www.amazon.co.uk/Global-Gizmos-5-inch-Benross-Gadget/dp/B00DW0GLDA/ref=pd_lpo_21_img_1/257-2719001-5808939?_encoding=UTF8&pd_rd_i=B00DW0GLDA&pd_rd_r=395b98c1-9a34-4bd8-8db4-d8579887ef8f&pd_rd_w=KVqXl&pd_rd_wg=cgMPq&pf_rd_p=7b8e3b03-1439-4489-abd4-4a138cf4eca6&pf_rd_r=ME287NC6MRTPVGC338GV&psc=1&refRID=ME287NC6MRTPVGC338GV. Aug. 2020.

[2] K. Antonakoglou et al. "Toward Haptic Communications Over the 5G Tactile Internet". In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 3034–3059.

[3] Dr. Luca Brayda. *Personal Assistive Device for BLIND and visually Impaired people*. Project Final Report FP7 - 611621. Version 2017-09-30. Fondazione Istituto Italiano di Tecnologia, Sept. 30, 2017.

[4] Physik Instrumente USA - Precision Motion Control. *Ultrasonic Motor Operating Principle PILine Linear / Rotary Piezo Motor by www.pi.ws*. Youtube. 2015. URL: https://youtu.be/9F8qU5LaHXY.

[5] M. Eid, j. Cha, and A. El Saddik. "Admux: An adaptive multiplexer for haptic-audio-visual data communication". In: *IEEE Trans. Instrum. Meas.* 60.15 (2011), pp. 21–31.

[6] Scharon Harding. *Hands-on With Razer's 8,000 Hz Gaming Mouse: Avalon Prototype Claims to Be 8 Times Faster*. tom's Hardware. Jan. 10, 2020. URL: https://www.tomshardware.com/news/hands-on-with-razers-8000-hz-gaming-mouse-avalon-prototype-claims-to-be-8-times-faster.

[7] *High-definition haptics: Feel the difference!* slyt483. Texas Instruments. Sept. 2012.

[8] JLCPCB.COM. *PCB Capabilities, Know JLCPCB's Capabilities & Get your PCBs Built Fast*. JLCPCB. 2020. URL: https://jlcpcb.com/capabilities/Capabilities.

[9] Hiroyuki Kajimoto. "Electro-tactile Display: Principle and Hardware". In: Apr. 2016, pp. 79–96. ISBN: 978-4-431-55771-5. DOI: 10.1007/978-4-431-55772-2_5.

[10] Hiroyuki Kajimoto. "Electro-tactile Display: Principle and Hardware". In: *Pervasive Haptics: Science, Design, and Application*. Ed. by Hiroyuki Kajimoto, Satoshi Saga, and Masashi Konyo. Tokyo: Springer Japan, 2016, pp. 79–96. ISBN: 978-4-431-55772-2. DOI: 10.1007/978-4-431-55772-2_5. URL: https://doi.org/10.1007/978-4-431-55772-2_5.

[11] Hiroyuki Kajimoto, Masaki Suzuki, and Yonezo Kanno. "HamsaTouch: Tactile vision substitution with smartphone and electro-tactile display". In: *Conference on Human Factors in Computing Systems - Proceedings* (Apr. 2014). DOI: 10.1145/2559206.2581164.

[12] Seung-Chan Kim et al. “SaLT: Small and lightweight tactile display using ultrasonic actuators”. In: Sept. 2008, pp. 430–435. ISBN: 978-1-4244-2212-8. DOI: 10.1109/ROMAN.2008.4600704.

[13] Jelger Lemmers. *Haptic Demo*. TU Delft. Aug. 12, 2020. URL: https://www.youtube.com/watch?v=krDbwzEhgUc.

[14] Manus. *Manus-VR*. Manus-VR Prime Haptic. Jan. 15, 2020. URL: https://manus-vr.com.

[15] J. Martínez et al. “Identifying Virtual 3D Geometric Shapes with a Vibrotactile Glove”. In: *IEEE Computer Graphics and Applications* 36.1 (2016), pp. 42–51.

[16] Learning Resources. *Improve 3D Print Quality by Preventing Those Pesky Blobs!* Simplify3D. 2016. URL: https://www.simplify3d.com/preventing-blobs-on-3d-print/.

[17] Senso. *Senso glove*. Senso. Jan. 15, 2020. URL: https://senso.me/.

[18] Anders Södergård and Mikael Stolt. “Properties of lactic acid based polymers and their correlation with composition”. In: *Progress in Polymer Science* 27.6 (2002), pp. 1123–1163. ISSN: 0079-6700. DOI: https://doi.org/10.1016/S0079-6700(02)00012-6. URL: http://www.sciencedirect.com/science/article/pii/S0079670002000126.

[19] Gernot Sümmermann. *Cynteract glove*. Cynteract. Jan. 15, 2020. URL: www.cynteract.de.

[20] *Tactile Feedback Solutions Using PiezoelectricActuatorsr*. 4706. Maxim Integrated. Jan. 2011.

[21] Y. Tanaka, Y. Goto, and A. Sano. “Haptic display of micro surface undulation based on discrete mechanical stimuli to whole fingers”. In: *Advanced Robotics* 31.4 (2017), pp. 155–167. DOI: 10.1080/01691864.2016.1262790. eprint: https://doi.org/10.1080/01691864.2016.1262790. URL: https://doi.org/10.1080/01691864.2016.1262790.

[22] Textures.com. *Texture Database*. CGTextures. Jan. 12, 2021. URL: https://www.textures.com/.

[23] Textures.com. *Textures.com Terms and Conditions*. CGTextures. Jan. 12, 2021. URL: https://www.textures.com/terms-of-use.html.

[24] D. Van Den Berg et al. “Challenges in Haptic Communications Over the Tactile Internet”. In: *IEEE Access* 5 (2017), pp. 23502–23518.

[25] V. Vechev et al. “TacTiles: Dual-Mode Low-Power Electromagnetic Actuators for Rendering Continuous Contact and Spatial Haptic Patterns in VR”. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. Mar. 2019, pp. 312–320. DOI: 10.1109/VR.2019.8797921.

[26] Ramiro Velázquez, Edwige Pissaloux, and Michael Wiertlewski. “A compact tactile display for the blind with shape memory alloys”. In: vol. 2006. June 2006, pp. 3905–3910. DOI: 10.1109/ROBOT.2006.1642300.

[27] David Vokoun et al. “Magnetostatic interactions and forces between cylindrical permanent magnets”. In: *Journal of Magnetism and Magnetic Materials* 321.22 (2009), pp. 3758–3763. ISSN: 0304-8853. DOI: https://doi.org/10.1016/j.jmmm.2009.07.030. URL: http://www.sciencedirect.com/science/article/pii/S030488530900746X.

[28] Sang-Yeun WON et al. “Two-point discrimination values vary depending on test site, sex and test modality in the orofacial region: A preliminary study”. In: *Journal of Applied Oral Science* 25 (Aug. 2017), pp. 427–435. DOI: 10.1590/1678-7757-2016-0462.

[29] Juan Zarate et al. “Keep in Touch: Portable Haptic Display with 192 High Speed Taxels”. In: May 2017, pp. 349–352. DOI: 10.1145/3027063.3052957.