VRIJE UNIVERSITEIT BRUSSEL

Graduation thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Applied Sciences and Engineering: Applied Computer Science

TOWARDS INVESTIGATING THE DESIGN SPACE OF DATA PHYSICALISATION

AHMED KAREEM A ABDULLAH

Academic year 2017–2018

Promoter: Prof. Dr. Beat Signer
Advisor: Payam Ebrahimi
Faculty of Engineering

Abstract

Data physicalisation is a new research area that spans Information Visualisation (InfoVis) and Scientific Visualisation (SciVis) [117]. However, there are many other disciplines which have been actively researching for novel methods to represent and explore data in order to keep up with the growth in data and its forms. For instance, the field of human-computer interaction endeavours finding new forms of interaction with data beyond the traditional mice and keyboards. For instance, researchers in the field of Tangible User Interfaces (TUIs), have been focusing on coupling the physical and digital domains into one coherent system. Eventually, this provides users an immersive experience while exploring data by affording direct interactions with the underlying models through common physical artefacts. Nevertheless, the research interests provide new aspirations to find interaction terms beyond the static physicalisation in tangibles. We also have some visionary work such as the Radical Atoms proposed by Ishii et al. [49] promising a new horizon of interactions considering the possibilities allowed by the programmable matter, which is a dynamically reconfigurable material of the future. The WISE lab is also involved in the design-driven research towards enabling programmable matter. Researchers in the WISE Lab explored the idea of simulating programmable matter and investigated some important concerns to the development of a system with such a capacity. The tangible hologram (TangHo) platform is a result of that investigation, trying to best approximate the potential of programmable matter. Data physicalisation has many advantages and benefits, such as allowing intuitive interactions that are not learned conventions, providing immersion during the exploration, represent data in its original physical form and also encode data in terms of physical variables. Therefore, data physicalisation enables a natural understanding of data based on human's perceptional skills. One of the challenges of data physicalisation is finding the physical variables. These variables are crucial in order to apply data physicalisation and to evaluate its usability, as well as providing a basis that researchers can lay on to enhance insights in their vision-driven research about how to design for a programmable matter and future interfaces.

Developing evidence in Computer Science depends on a good theoretical foundations for developing technologies or methods, as well as upon its experimental validation. Therefore, in this thesis, we investigate the design space of data physicalisation in order to provide an elementary model of physical variables. The thesis surveys the type of interaction in a broad set of existing work to find a common pattern in designing physicalisation and multisensory representations in relation to the underlying data. This investigation establishes a baseline for deeper research to find the taxonomies of physical variables in data physicalisation. Moreover, we introduce the thermo-tactile system as proof of concept prototype for two of the resulting parameters. Finally, we perform an experiment to evaluate the effectiveness of data physicalisation and to validate our prototype.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my promoter Prof.Dr. Beat Signer for the continuous support that he provided during my study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. Therefore, I would like to thank him from deep inside my heart.

Besides my promoter, I must also express my very profound gratitude to my parents and to my girlfriend Dagmar for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. The thanks are also extended to my friends who helped me during the thesis.

Finally, I would especially like to thank Payam for his advises, encouragement and help which has a great contribution to me accomplishing my thesis. I would also like to thank all the members of the WISE lab for their kindness throughout the year. I wish everyone in the lab great success in all their career.

Contents

1
Introduction

Data physicalisation is an emerging field of research concerned with the representation, utilisation, interaction, design and analysis of data in physical forms. Representing data in physical forms has great advantages and benefits, such as recruiting human perceptual skills in the data exploration process, conveying the information extracted from the data into multiple channels by utilising intermodal human perception, such that a user can perceive multiple aspects of data at the same time. Nevertheless, such a representation does not depend on the visual sense only as the dominant channel in representing data or what is known as visualisation. Visually-impaired people can be helped to perform analysis tasks and get insights from the explored data through other modalities [10]. However, it also has some challenges such as how to convey data effectively through multiple modalities? What type and form of interactions are possible with such data? How to expand the application domain with more applications? What are the suitable evaluation methods for physicalisations? [55]. From this preselective, modelling the design space of data physicalisation attracts the most interests in the research. Moreover, data physicalisation as a field has some relation and drawbacks to other fields, such as information visualisation and human-computer interaction (HCI). The relation to the former is by means that both fields seek new methods to represent data, eventually facilitating the exploration process to users, whereas, the relation to the latter is that both fields are developing new forms of interactions beyond the traditional ones. Therefore, investigating those fields as well can speed up the process of finding the design space features of data physicalisation as one can learn from similar methods and practices that have already been found for visualisations and HCI.

This thesis aims to investigate the design space model of data physicalisation as well as the aforementioned related fields. In order to find a common patterns between data and what could be seen as their physicalisation, based on a set of current examples mainly from the field of human-computer interactions. The investigation can also help in the future development of the tangible hologram platform (TangHo) [97]. The platform is concerned with exploring physical representations of data, aiming to provide the users with an immersive exploration experience. Therefore the implementation and interaction challenges associated with data physicalisation also comply with TangHo, as TangHo is a platform for data physicalisation. Also as a result of this investigation, we propose a prototype of a new physicalisation interface for thermal and tactile feedback. This prototype forms a computer-supported physical representation of data, which is in line with the definition of data physicalisation [55]. We then also experiment with the prototype to evaluate the effectiveness of such a physicalisation. In the remainder of this chapter, we introduce our problem statement, our solution and its contribution to the field, and the methodologies that have been followed in order to accomplish our goal.

1.1 Problem Statement

In their research agenda, Jansen et a.l [55] highlighted some opportunities and challenges for data physicalisation. The main highlights were about how to encode data physically in an effective manner and how to support the interaction with such data. However, these two main challenges are then split into several puzzles; their aggregation will lead to forming proper design guidelines. As the focus of data physicalisation is on making data physical, understanding the design space of these physical representations is one of the main challenges [55]. Similarly, finding the effectiveness in physicalising data and their design implementation, Jansen et al. [55] proceed by noting a common challenge between data physicalisation and information visualisation, as how to effectively transform the information from the data domain (i.e. raw data) to the representation domain, "*Data Physicalisation and Visualization share the problem of finding suitable transformations from digital data to human readable representations.*" [55]. Usually, in information visualisation, this transformation happens utilising *visual encodings*, which is the process of mapping data points to *visual marks* (e.s. points and lines) and data attributes to visual variables (e.s. position, size and length) [6]. The basics of this mapping process together with other guidelines have been developed for some time by Jacques Bertin in his astonishing work "*Semiology of Graphics*". However, in data physicalisation, we need to have similar principles in order to create meaningful physicalisation that are not just a result to purely artistic data sculptures. Jansen et al. promote the importance of such variables for data physicalisation, as "*identifying, exploring, and classifying physical variables is a research challenge that will be key to understanding the design space of data physicalisations*" [55]. Moreover, this conviction has been supported by other researchers before Jansen, but it was not directed towards data physicalisation explicitly. For example, Loftin pointed a similar challenge in deploying a multisensory representation platform. Loftin stated that "*one major barrier remains: a knowledge of what type of variable maps best on to which sensory channel and in what way*" [64]. Also, Roberts and Walker made a similar call but for perceptual variables, "*to work towards a holistic theory of perceptual variables we need to understand the capabilities and limitations of each variable of each sense*" [90], as their work was about using all human sense to represent data, which they define as multisensory information visualisation. Although, multisensory data representations are not precisely data physicalisations, however, they share similar principles, such as, convey data in a multimodal manner and use human perceptual capacity to understand the data intuitively.

Although physical representations of information existed already since ancient times through the embodiment in data sculptures (e.g. sculptured tokens), however, the literature work in the data physicalisation research area is limited. This lack of studies in this field can be interpreted as a result of the novelty of the field itself and the number of active contributors involved in. However, that does not affect the continuity or quality of the research. On the contrary, it opens up many opportunities for researchers and encourages innovation and creativity. Also, the fields associated with data physicalisation, posses a considerable amount of literature work. Thus one may use this inventory to develop a better understanding of data physicalistaion.

Based on the presented problem statement and its importance, the objectives of this thesis can be formed as follows:

- **To investigate** the possibility of using existing work in related fields for the benefit of a better understanding of the data physicalisation design space.

- **To model** a basic baseline towards a deeper investigation of physical variables.
- **To develop** a data physicalisation prototype based on the proposed model.
- **To evaluate** the effectiveness in data physicalisation using the prototype.

1.2 Contributions

The physical variables are crucial blocks for the design of data physicalisation. Therefore, exploring and identifying these variables is an interesting and challenging research question, which is also a focus of this thesis as explicitly stated in the objectives. Moreover, researchers mostly refer to Bertin's work [6] when trying to support the importance of encoding guidelines. Also, others refer to the same work when trying to clarify the principle of encoding behind their work [53, 54, 75]. However it is evident that there is no such work in the field of data physicalisation. Moreover, initiating a research area with a goal of finding similar guidelines to the ones of Bertin will be a great contribution to the field of data physicalisation and to HCI in general, considering the common concerns between both fields. However, in his work Bertin did holistic research into the semiotics of graphics in order to create a general guideline for the design. Moreover, in his research, he illustrates the relationship between graphics and the way human perceive them, based on scientific reasoning supported by examples from psychology, as well as exploring common practices used to generate graphs, together with a set of artistic works. Therefore, it is not possible to obtain such a comprehensive outcome within the time frame of this thesis. However, the research with the aim of reaching a similar outcome as in Bertin's work will be divided into several parts that can be covered by some studies. Thus, the general contribution of this thesis based on its goals can be stated as follows:

- **A comprehensive survey and classification of the state of art examples, together with the most relevant guidelines and frameworks that have been made in HCI**. Besides its importance to understand the design space of data physicalisation, this survey will help to those working in the field of human-computer interaction as a reference that summarised the current development in the field.
- **The exploration of common patterns between the data type and the interaction method as a first step in the exploration**

of physical variables. This exploration can be a great contribution to both HCI and data physicalisation. It contributes to HCI, as most of the work in HCI focuses on several aspects regarding the interaction itself, but not the data type being handled by this interaction. On the other hand, it contributes to data physicalisation, as it clearly lists a set of possible technologies and practices that can be used to design the interactions with physicalisations. Moreover, these interaction methods are collected from different modalities, to form a good reference to those wishing to develop multimodal data physicalisation interface.

- **The design and development of a multimodal interface for physical data representation.** The interface combines thermal and haptic feedback as channels to convey data, eventually representing data physically in the form of heat and tactile feedback. Moreover, the production of the prototype is made by using low-cost and widespread equipment, which makes it easily reproducible.

- **A new application domain for data physicalisation.** Based on the work undertaken in this thesis and the experiment performed using the produced prototype, we foresee a great potential for data physicalisation in representing different data types and formulate a new research question on how to physically represent multidimensional data.

1.3 Methodology

The methodology of this thesis conforms to the Design Science Research Methodology (DSRM) by Peffers [84] and its complementary work by Offermann [79]. The DSRM is generally meant to meet three objectives; consistency with prior literature, providing a structure of process for doing the research and providing a logical model for presenting and evaluating the research [79]. These objectives are then implemented through six phases; problem identification and motivation, objectives definition, design and development, demonstration and communication [84]. In this thesis, the problem is well identified, and its value is described in the problem statement section, which also supports the motivation of the work undertaken in this thesis. Moreover, the objectives of a solution are also described, and those define the ultimate goal of our research, which is to identify the physical variables and to come up with guidelines similar to Bertin's work [6]. However, it is also explicitly mentioned what is feasible for this thesis to achieve as centred objectives. Moreover, clarifying the potential contribution of this thesis, as presented in the previous section, together with what is just mentioned, constitutes a full definition of the solution's objectives. Furthermore, the design and development of an artefact are achieved in two steps, first theoretically in Chapter 4 where the survey and the primary classification is and second in Chapter 5 where the prototype implementation is discussed. However, it is also important to mention that Thematic Analysis is used as a methodology to approach and accomplish the conceptual modelling in Chapter 4. Thematic analysis (TA) is a widely-used qualitative data analysis method. It is "*one of a cluster of methods that focus on identifying patterned meaning across a dataset*" [118]. Nevertheless, the description and discussion of the thematic analysis will follow in Chapter 4, whereas, Chapter 6 fits the demonstration phase of DSRM through a simple experiment using the prototype. Finally, the communication phase is ongoing, and we have a clear plan to publish the outcomes of this investigation.

1.4 Thesis Outline

Since this chapter introduces the reasons and goals behind the work undertaken in this thesis, Chapter 2 covers the necessary background to put some context to this thesis. In there, the data physicalisation, its evolution and its benefits, as well as the related fields to it are explored. In Chapter 3, work related to our investigation is described and a comprehensive description of the survey together with the classification is presented in Chapter 4. Also, the discussion of each work and the reasoning behind its classification is illustrated in Chapter 4. Chapter 5 then provides a detailed description of the implementation of the prototype. This includes implementation details and illustrations for hardware, software and the design of the interface. Chapter 6 discusses the design and results of the experiment made by using the prototype. Here we present the dataset used for the experiment, as well as the setup and methods used to obtain the results. Finally, potential research goals and a summation are explored in the last chapter.

2
Background

Data physicalisation as coined by Jansen et al. [55] considers multidisciplinary research domains. These are trying to form new methods for representing and interacting with data. In the same matter, data physicalisation as a field has some roots in other scopes forming its multidiscipline nature, such as human-computer interaction and its subdomain of tangible interaction, information visualisation and data representation in general. This chapter provides a brief introduction to the context of this thesis.

2.1 Human-Computer Interaction

Human-Computer Interaction (HCI) is a multidisciplinary research area interested in all the factors that concern the communication between users (human) and machines (computer). In their book Human-Computer Interaction Dix et al. [26] define HCI as "*the study of the way in which computer technology influences human work and activities*". The fundamental goal of HCI is to facilitate the interactions between users and computers. This is achieved by applying the *interaction design* rules of HCI which are serving to produce a system that mimics some *usability goals*. In other words, "*designing products that are easy to learn, effective to use and provide an enjoyable user experience*" as stated by Rogers et al. [91]. Dix et al. also support this vision when they explain the relation between HCI and other

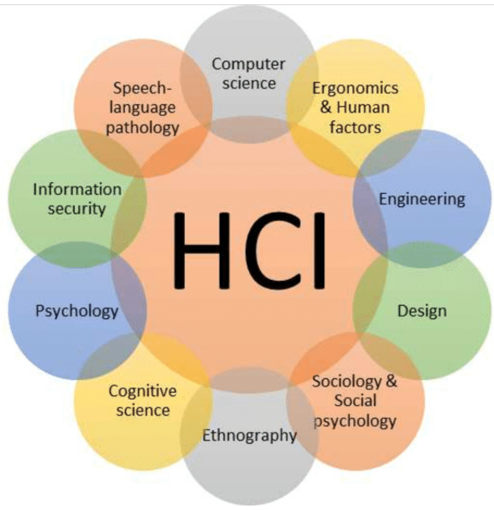


Figure 2.1: Human-computer interaction and related research fields © Chakraborty et al.[18]

design disciplines "*HCI has an associated design discipline, sometimes called Interaction Design [...], focused on how to design computer technology so that it is as easy and pleasant to use as possible. A key aspect of the design discipline is the notion of (usability) which is often defined in terms of efficiency, effectiveness and satisfaction*" [26]. The research field of HCI started in the early 1980s since the term Human-Computer Interaction is familiarised by Card et al. in their book '*The Psychology of Human-Computer Interaction*' [15]. By then, it was more specialised in Computer Science covering cognitive science and human factors. However, nowadays HCI aggregates much more beyond Computer Science, since different fields are involved in constructing the ultimate goal of HCI. Figure 2.1 illustrates the multidisciplinary nature of HCI. Nevertheless, HCI opened a wide space of research, which in turn helped to improve the way that users used to perform their tasks and go beyond the simple use of mouse and keyboard. Researchers are actively providing recommendations and guidelines for designers about best practices and how a good interactive system should be built, such as the ones for *multimodal interaction* [81] in terms of the design [76], usability [22] or, integration [76, 45]. Simultaneously, the research in this field resulted in creating new sorts of interfaces that go beyond the graphical user interface which pours in enhancing the user experience as a goal [91], including graspable user interfaces [29], the tangible user interfaces [50] or multimodal interfaces [28].

2.1.1 Tangible Interaction and Tangible User Interface

Tangible Interaction is a generalised term for graspable user interfaces [29], tangible user interfaces (TUIs) [50] and embodied interaction [27]. The early appearance of this type of interfaces was in the work of Fitzmaurice et al. [29] when they introduced the concept of the graspable user interface. The definition back then was about physical handles called bricks "*that allow direct control of electronics or virtual objects*" [29]. Those physical handles act like input devices that are "*tightly coupled or attached to virtual objects for manipulation or for expressing action (e.g. to set parameters or for initiating processes)*" [29]. The first prototype introduced the *ActiveDesk*, where bricks are used on top of a large horizontal display surface. However, in their work on *Tangible Bits* [50] Ishii et al. went beyond this definition, establishing a more general approach known as tangible user interfaces (TUIs). The definition of tangible bits is that it "*allows users to grasp and manipulate bits in the centre of users attention by coupling the bits with everyday physical objects and architectural surfaces.*" The main idea of TUIs is to go beyond the traditional graphical user interfaces (GUIs) by augmenting the physical space through the coupling of digital information and physical objects and environments. In other words, transforming the relation from being between bits and pixels to a "*coupling of bits and atoms*" [50], which in turn provides new forms of input rather than a mouse and keyboard mediating the interaction with GUIs. The traditional way of interaction through mouse and keyboard are learned conventions, whereas it is a direct relationship between physical objects and the underlying data in TUIs. More clearly, the affordance of physical objects is understood intuitively by the human. Thus, their affordance can illustrate the use of the interface and the set of the possible interactions [48]. As prominent contributors to the field of tangible interaction, Ishii and Ulmer followed up their work on tangible bits. In their work on *Emerging Frameworks for Tangible User Interfaces*, they set the fundamentals and guidelines for designing TUIs. In addition to that and by using similar approaches from software development, Ishii and Ulmer introduced the model-control-representation (*MCRpd*) 2.2B (physical and digital) architecture based on the model-view-controller (*MVC*) 2.2A archetype. The definition of the view is altered to two subcomponents. These are *physical representations '(rep-p)'* for the physical components embodied in the artefact and *digital representation '(rep-d)'* for the computational components of the interface that are not physical (e.g. video projection). The model and the controller are extended from MVC. However, they also describe a form where the controller is embedded with the physical artefact. This design pattern can be extended to the data physicalisation scope since one can use a similar

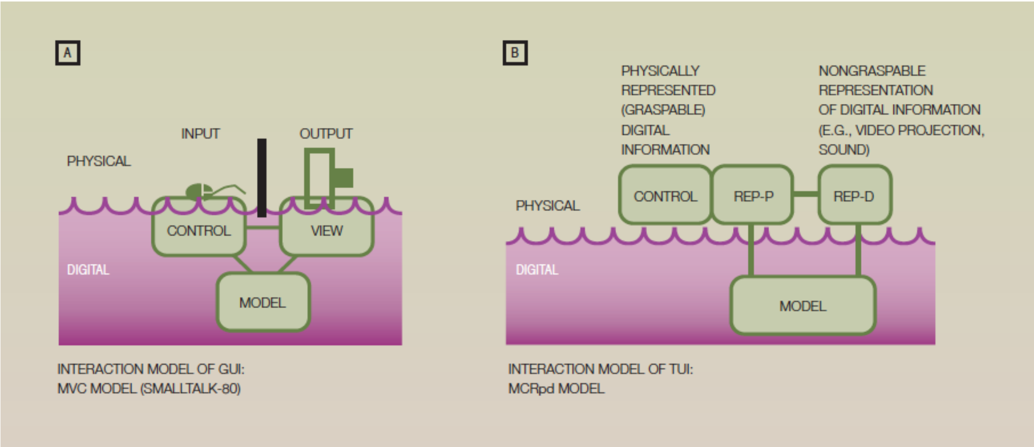
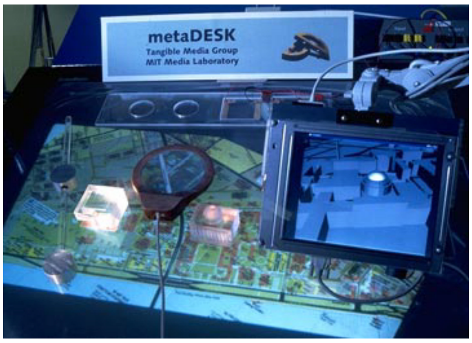


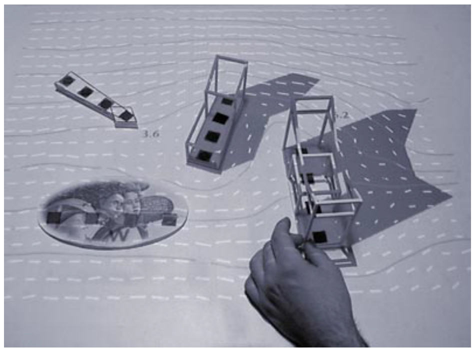
Figure 2.2: MVC and MCRpd © Ullmer and Ishii [113]

approach when designing a computer-supported data physicalisation prototype. Ishii and Ulmer also provided some models and prototypes for TUI to support the understanding of the design, including *metaDesk* [114] and *URP* [113]. The metaDesk shown in 2.3a platform is the very first example of a TUI, where the idea was to step away from a traditional GUI towards a similar interface but as a TUI. The platform consists of a back-projected graphical surface equipped with various tools. These tools are meant to reflect similar items as known from the GUI. For example, an icon is represented as a*phicon* [1], window with lens, handle with(*phandle*) [2] and widget as instrument. The use case of the platform was to allow users to manipulate graphical maps of the MIT campus. Phicons were coupled to the digital information of buildings on the campus. By placing a phicon on the surface, a graphical representation of the map will appear with the aligned location of the building that the phicon is linked to. By placing another phicon on the desk surface, the view changes in relation to both phicons. A zooming operation is also possible by pushing or pulling the phicons closer or farther away from each other. Moreover, rotating one phicon while keeping the second one fixed will result in a rotation of the whole map. This last operation can also be achieved by using some dedicated physical instruments. Similarly, URP shown in Figure 2.3b is an urban planning system, which simulates environmental factors such as wind and the shadows in relation to the geospatial distribution of the surrounding buildings. The URP platform remarkably depicts the integration of the physical models with the projected interactive

[1]referring to a physical icon, where 'p' stands for physical
[2]referring to a physical handle



(a) (b) metaDesk © Ulmer and Ishii [113]



(b) (a) URP © Ullmer and Ishii [113]

Figure 2.3: Tangible User Interfaces from MIT Media Labs

simulation. Any repositioning or reorientation of the physical objects (buildings) will result in a change in the simulation of the wind and the shadows. Moreover, the material of buildings can also be changed. A *material wand* such as glass can be used with the effects of the change being reflected in the simulation. Other physical tools are also available to, for example, measure the distance between two points or the wind speed.

Benefits of TUIs

From the aforementioned examples, it is clear that the interaction with interfaces feels more intuitive and natural. As previously mentioned, the affordance of the physical objects defines a perspicacity about the whole system. Moreover, the type, shape, colour, material, texture and all the other physical properties of the physical object can extend the traditional internal computer representation [51], because these elements can offer an extra layer to encode information. At the same time, these properties can shift to more specialised context-sensitive input devices [41]. An example is URP, where the material type affects the whole experience. Nevertheless, the size and nature of the interface is mostly built on a surface background allows for collaborative work between groups [44], such as in URP where multiple persons can experience the simulation at a given time. This also concerns navigation in the environment as an interaction modality where the movement is used to explore the model, as one can walk around the workbench to get different perspectives. In addition, it is evident that such types of interfaces attract users and provide them a more enjoyable experience [44]. Moreover, Hornecker and Shaer elaborate different social and psychological factors that can be enhanced by representing the scope of an idea via tangibles [44, 96].

Limitations of TUIs

Ishii and Ulmer also discussed the current limitation of TUIs, as they are static physical embodiments with limitations to expose changes in their material or physical properties. Therefore, they are limited in reflecting dynamic changes to the underlying data, when the underlying models afford such changes. Ishii and Ulmer illustrated this limitation through a simple example of comparing pixels on screens with tangibles. Pixels afford to change the form, position or even the properties of an object (e.g. colour and size), whereas it is not the case with tangibles. Thus, tangibles are normally not able to express variant physical properties at the same time within the same artefact, which can make them *inconsistent with underlying digital models* [48] and ultimately result in an insufficient representation of the data to users. Furthermore, the most common approach in designing TUIs is the tabletop TUI or tangible workbench such as in URP. In such interfaces, it is essential to have "*a balance and strong perceptual coupling between tangible and intangible (dynamic) representations*" as stated by Ishii and Ulmer where dynamic representation refers to the video projection. Such a balance is important because if one relies too much on the digital representation (e.g. video projection), the whole interface will appear like an extended GUI with some extra controllers (appart from mouse and keyboard), where the advantages of being tangible are lost. In many cases it is difficult to achieve such a seamless coupling as stated by Ishii and Ulmer: "*In many successful TUIs, it is critical that both tangible and intangible representations are perceptually coupled to achieve a seamless interface that actively mediates interaction with the underlying digital information and appropriately blurs the boundary between physical and digital*". Finally, there are also some difficulties in keeping the digital and the physical state of the interface synchronised. These are mostly due to the technologies that the tangible interface are made of.

However, these constraints in tangibles are due to the affordances of the artefacts themselves in terms of material, shape and physical properties or a limitation in the existing technologies when creating a TUI. Despite these limitations, these techniques still have a significant contribution to HCI and different applications are still employing the TUI nature based on the discussed guidelines and frameworks. For example, Spindler et al. [102] used a similar approach as in the metaDesk by utilising a passive display (magic lens) that can provide a three-dimensional exploration of a dataset. Similarly, Hogan et al. [41] designed a cube to represent a real-time data level of hydrogen in the deep space as complementary to traditional on-screen visualisation. The cube is embedded with vibration feedback to give an idea about the intensity of hydrogen at the moment that the user shakes the cube. Moreover, Jansen et al. [51] used different tangibles as examples to similar desktop visualisation, to represent a model for visualisation beyond the desktop. Another example is *MakerWear* by Kazemitabaar et al. [59], a wearable construction toolkit for children. This wearable toolkit of small plug and play tangibles allows children to create self-expressive and meaningful computational designs.

2.1.2 Human-Material Interaction (Radical Atoms)

Human-Material Interaction is an envisioned field of research coined by Ishii et al. [48] as a successor of the TUI. The general concept is that all the digital information will have a "*physical manifestation*" that is seen as a physical twin of the digital version of data, where the human can directly interact with and manipulate. Ishii et al.set their vision under the name *Radical Atoms*, which allows for a *vision-driven* research into a form of Human-material interaction. Moreover, Ishii et al. support their visionary idea by describing a hypothetical type of material they call *Perfect Red*. The hypothetical perfect red is a programmable physical material (*a digital clay*) that is "*computationally transformable and reconfigurable*". Such a generation of material must have the ability to change its form, appearance and the physical properties dynamically. Thus, such features allow having *bidirectional coupling* between the material (i.e. atoms) and the underlying digital models (i.e. bits). In other words, the material must be able to reflect dynamic changes bidirectionally in real-time (i.e. changes in the digital model are reflected in the physical material and vice-versa). Thereby, the dynamic affordance of the perfect red will behave as a medium for the representation and input interactions. Furthermore, Ishii et al. also proposed an MVC like architecture to define further the requirements for using such a material and to ensure its bidirectional nature as illustrated in Figure 2.4. The *transform* part depicts the ability of the material to transform its shape to modify the underlying digital model. This is when the user manipulates the material interface, for example through direct gestures with the clay.

The *conform* part is to provide constraints of the environment and to set the possible interactions that users can perform for input. These constraints are programmable, since such a material can radically change its shape and physical properties dynamically and in users' vicinities, they must follow such constraints which are mostly imposed by physical laws as stated by Ishii et al. (e.g. total volume that the material can expand to, should be smaller than the room size). The reason behind these constraints is to not have unexpected or unwanted behaviour, not in terms of reflection (representation), neither in term of interaction (input), such one can ensure users'

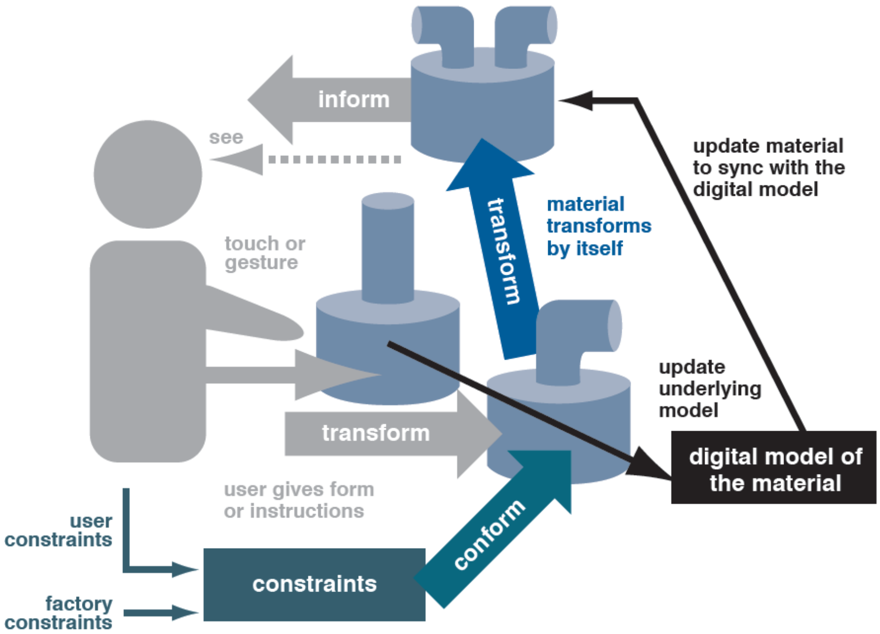


Figure 2.4: Interactions with Radical Atoms. © Ishii et al. [48]

safety all the time. Finally, the *inform* part describes the dynamic affordance of the material which serves for apprising users about the transformational capabilities of the interface. As these materials are dynamically changing during the interaction, the users have to be always informed about the state of the interface, and the possible interaction they can perform at the moment in relation to the state of the interface.

The vision of Radical Atoms is able to overcome the limitations of TUI described earlier. The materials underlying this vision are wholly and directly coupled with the digital information. Also, they have the must property to reflect dynamic changes and eventually their dynamic affordance allows for appropriate interactions by users. However, the vision is not meant to be a solution for current users' needs. Therefore, it is defined as vision-driven research, Ishii et al. illustrate that "*The reason why we focus on the vision-driven approach is its lifespan. We know that technologies become obsolete in about one year, users' needs change quickly, and applications become obsolete in about ten years. However, we believe the strong vision can last beyond our lifespan*". Indeed, the vision opens different research questions regarding the future materials that will enable the Radical Atoms vision. It also, raises a new research community in the field of interaction design as one has to investigate the blueprint of interactions with such dynamic affordances.

2.2 Data Physicalisation

Data physicalisation is a relatively new field of research which aims to improve the way that people used to explore, understand and analyse data, as well as the interactions associated with these processes. The research derived its orientation towards data physicalisation as a result of investigating the possibilities of new forms of representation or what called information visualisation beyond the desktop [51, 54]. However, data physicalisation as a term has been coined by Jansen et al. in their research agenda [55], pointing interests about data physicalisation and its advantages, effectiveness, utility and opportunities and challenges in physicalising the data. Jansen et al. propose the data physicalisation as a research area "*that examines how computer-supported, physical representations of data (i.e. physicalisations), can support cognition, communication, learning, problem-solving, and decision making*". Nevertheless, data physicalisation is named differently by different researchers. Such that Ogi and Hirose investigated the possibility of employing several senses to represent scientific data [80], describing a calibration method by using visual, acoustic and touch system. Similarly, Nesbitt tried in several investigations to model the design space of such multisensory representations [74, 75]. Nevertheless, Loftin in his work also described a new multisensory system; he defined as *a scientific-engineering-workstation* [64], which also aims to use multiple sensory channels as simultaneous displays. Moreover, Roberts and Walker [90], Obrist et al. [78], made the same calls in their works, aiming for multisensory information representations. Hogan and Hornrcker [42] made a comprehensive investigation into many domains trying to form a design space for what they call multisensory data representation. However, the term data physicalisation happens to be more general than all previous terms, since it is intended to cover all aspects related to the representation and interaction with data. However, these different naming share relatively the same interests, which is facilitating users' exploration and interaction tasks with data. Therefore, we consider the work underlying these names related under the general term of data physicalisation.

Jansen et al. define physicalisations as *data-driven* physical artefacts, which are physical counterparts of data visualisations [55]. Precisely, they define data physicalisation or simply physicalisations as "*a physical artefact whose geometry or material properties encode data*". Such artefacts may involve the use of computers, either to fabricate them or to actuate them, as illustrated by Jansen et al. under the term of *data sculpture*. The term data sculpture is older than data physicalisation, and it generally refers to artistic data physicalisations. In 2008, Zhao and Moere [124] defined the embodiment in data sculpture as "*a data-based physical artefact, possessing both artistic and functional qualities, that aims to augment a nearby audience's understanding of data insights and any socially relevant issues that underlie it*". However, Jansen et al. generalised this term as "*data-driven artefacts [...] built by artists and designers who seek to elicit emotions and convey meaning beyond mere data*". Nevertheless, the data representation was depicted in a physical form in the ancient time, even before the invention of papers or screens, via artefacts such as the abacus. Technological advances in 3D printing and shape-changing displays offer the opportunity to bring data back to the physical form in creative and innovative ways that aim to enhance users' exploration and interaction tasks in a purposeful manner.

Relevance of Visualisations and HCI to Data Physicalisation

From the definition and the goals of data physicalisation, it is clear that there is a close relation to the field of information visualisation as well as to the field of human-computer interaction [2].

The relation to the former is by means that both fields concern data representations, and eventually facilitate the exploration process to users. Data physicalisation can offer some advantages over the traditional visualisation, as the physicalisations can extend different channels to encode data, by employing human senses or any range that depends on human perception. For example, haptic, visual, olfactory, auditory senses, as well as using weight, inertia, texture, density, size, temperature and much more physical variables to encode data. These variables will allow extending the dimensions of the representations that are currently restricted to a set of visual encodings. However, finding the physical variables and the useful mapping between them and data dimensions is one of the major issues in the field of data physicalisation. When discussing the opportunities and challenges in data physicalisation, Jansen et al. stated that "*Identifying, exploring and classifying physical variables is a research challenge that will be the key to understanding the design space of data visualisations*". Although, this issue is of vital importance to the field, not much work has been done in this matter. Jansen and Hornbaek [52] made a psychophysical investigation in the use of size as a physical variable and they concluded that "*the use of perceptually-optimised size scales seems possible*". However, they also call for more empirical studies on the size and other physical variables, in order to find a standard approach for the encodings between data points and physical variables. Moreover, the evidence about the importance of such physical variables is also supported by Jansen et al. when they described the relation between data physicalisation and visualisations, as well as the common problem that both fields consider [55]. As both fields trying to find a suitable transformation from the digital domain (i.e. data) to the representation domain (i.e. visualisations, physicalisations). In information visualisation, this transformation happens to utilise *visual encodings*, which is the process of mapping data points to *visual marks* (i.e. points and lines) and data attributes to (visual variables) (i.e. position, size, length) [6]. The basics of this mapping process together with other guidelines have been developed for some time by Jacques Bertin in his astonishing work "*Semiology of graphics*". However, in data physicalisation, we need to have similar principles in order to create meaningful physicalisation that are not just a result to purely artistic data sculptures.

The relation to the later is because that both are seeking and developing new forms of interactions beyond the traditional ones. Jansen in her PhD dissertation [53] defined some relations between physicalisation, human-computer interaction and eventually human material interaction, such that one can adopt these interaction devices to allow for physical representations. As stated by Jansen, "*input devices can be specialised and adapted so that their physical shape reflects their functionality within the system; computer displays can be substituted by transformable shape changing displays or, eventually, by programmable matter which can take any physical shape imaginable*". Moreover, Ishii et al. also describes a relation between tangibility and physicalisation as "*tangible design expands the affordance of physical objects so they can support direct engagement with the digital world*" [48]. Nevertheless, Signer and Curtin [97] discussed several challenges considering physicalisation of data and the interactions around it. Also, they proposed the tangible hologram platform (TangHo), as a solution for those trying to expand the development of physicalising the data, as well as trying to simulate the vision of Radical Atoms. Curtin in his thesis illustrates the contribution of TangHo to the field of data physicalisation as well as to the field of interaction design, as *this platform will be beneficial to those wishing to develop, prototype and test innovative ways to physicalise data in an uncomplicated and rapid fashion. Indeed, data physicalisations can be easily approximated without incurring the monetary costs that may apply when adopting other approaches. Interaction designers, in designing for interactions with physicalisations, not yet realisable given the limitations of today's technology, can use the tangible hologram system to move beyond the storyboard and explore and evaluate interactions with simulations of such physicalisations.*

2.2.1 The Benefits of Data Physicalisation

Although the field of data physicalisation is relatively new, a number of studies have been undertaken, such as investigating the effectiveness and benefits of data physicalisation [55], as well as the interactions with such physicalisations [106, 107], which resulted in a promising future to the field. Nevertheless, data physicalisation has many advantages, such as providing rich, compelling and intuitive interactions by utilising human's natural perceptual exploration skills, including active and depth perception. Regarding active perception, Jansen et al. describe that physicalisation can better exploit our active perception, such that "*a physical object like a hand-sized physicalization can be visually inspected by turning it around, by moving it closer, or by taking it apart. A large-scale physicalization can be explored by walking around. In contrast, on-screen visualisations need to explicitly support active perception by coupling input with output devices*" [55]. Moreover, Jansen et al. illustrate that physicalisations can provide better *spatial perception skills*, since the physical objects have richer details than computer displays, so that "*3D data can be perceived with less effort and more accuracy*". Moreover, physicalisation can utilise all senses to encode the information, each with their unique characteristics, so that the physicalisation is not limited to the visual sense. Therefore, physicalisations can convey data in different channels, enabling an intermodal approach for multisensory perception. Similarly, data physicalisation promotes the accessibility of data, such that visually-impaired people can make use of other modalities to explore and analyse data. Nevertheless, representing data in physical forms has other advantages for cognition and learning tasks, such as physical artefacts that are used in classrooms for mathematics (e.g. Geometric shapes), chemistry (e.g. Covalent bonds) and physics (e.g. Crystal molecules). Although, these physical artefacts are not physicalisations by nature and not meant to convey data,"*their cognitive and educational benefits will transfer to physicalisations. This is especially true for rearrangeable and interactive physicalisations*" [55]. This has been materialised by Chen et al. [19] as they used an innovative physicalisation approach for teaching abstract data analytical concepts, such as dimensionality reduction. Moreover, other researchers evaluated the memorability of physical representations [105], using multivariate bar charts to compare the memorability between the visualization and physicalisation. As a result, they claim that physicalisation led to significantly less information decay within the time span. Finally, physicalisation can be used for aesthetic and casual purposes, as well as to engage people with the presented data.

3
Related Work

As illustrated in the previous chapters, the goal of this thesis is to investigate in the desgin space of data physicalisation. Therefore, it is important to consider any work that has relations to the design space of the data physicalisation. Since one can shorten the way to a generalised design space model by looking at what has been demonstrated already, it may sometimes be confusing to see different terms called to the same concept, as data physicalisation has been approached differently by different researchers as explained already in the background chapter. Thereby, the discussion here will encapsulate all the terms under the concept of data physicalisation regardless of the different naming of different researchers. Nevertheless, investigating the design space of data physicalisation has some interests to the theoretical work that has been done in related fields. For example, the relation with HCI, specifically its subfield TUI and data physicalisation has been illustrated in the previous chapter, therefore, it may be interesting to look at the theoretical work in TUI, such as guidelines and frameworks created for the design of tangible interactions. This observation in turn can be useful if one decided to follow similar practices in forming terminologies for the design space of data physicalisation. Finally, in order to collect the new and most relevant papers about these disciplines, we examined different conferences such as the ACM Symposium on User Interface Software and Technology (UIST) [1],

[1] http://uist.acm.org/uist2018/

the ACM International Conference on Tangible, Embedded and Embodied Interaction (TEI) [2], the International Conference for Human-Computer Interaction (CHI) [3] and IEEE Visualisation (Vis) [4], as well as journal articles and papers from other domains such as physiology and psychology. In the following subsections we present the related work of the research areas mentioned above, in order of their importance in our investigation.

3.1 HCI

Fitzmaurice et al. described in their work, when they coined the term graspable user interface, the initial roadmap for the design space of graspable user interfaces based on the capabilities of the controllers or what they call bricks. Their guidelines designate multiple levels of details, which define the dimensions of the design space. Each dimension or level represents an abstraction that the designer should consider when constructing a new graspable interface. Thus, the focus in this work is on the characteristics of the bricks, their design, the type and the specification of the interaction through bricks, the communication between bricks and with the underlying digital model, the functionality provided, the specification of the environment and the surroundings where bricks function. Although the guidelines address the type of data coupling between the physical layer which is represented by the bricks and the virtual layer which is the underlying digital model, they have not touched on how the data should be represented or which data can be represented by what. Ishii and Ulmer went beyond that when they proposed the tangible user interface term in their work on Tangible Bits [50]. However, their research was concentrated on the general frame to design the interfaces themselves. The fundamental concept of the design was to enrich the user experience while interacting with the digital information. Therefore, the emphasis was on the use of the *interactive surfaces* and the use of the ambient media to achieve more immersive experience. Moreover, they deepened into the coupling between the physical tangibles and the digital worlds. Although these concepts have led to an evolution in the methods used in the preliminary graspable user interface and went beyond the traditional graphical user interface, they have not covered any data aspects. Since they were restricted to the use of the physical objects to manipulate data or *focused on the periphery of human perception* as stated by the authors [50]. Ullmer and Ishii [113] expanded the design space recommendations of the tangible

[2] `http://tei.acm.org/2019/`
[3] `https://chi2019.acm.org`
[4] `http://ieeevis.org/year/2018/welcome`

interfaces. The objective of their work was to group the existing frameworks in categories that can serve as a starting point for a border taxonomy. Although the work classifies tangibles based on the way of integration and configuration with the tangible interface, these observations provide richer insights into the relation between tangibles and the underlying data. For example, in what they define as a '*spatial*' category some examples have been used in "*visualisation-related capacities*" [113] such as representing geospatial data [114, 115]. In other categories, '*relational and constructive*' information about the attributes, operation or the simple structure of the data is presented. As mentioned, this classification helps improving the understanding of TUI design in relation to the representation intent by giving an overview of the mapping between tangibles and the underlying data. However, they were not explicitly generalised to cover which and how a particular data type can be represented through tangibles. Similarly, others have investigated the guidelines of tangible interaction more [116, 60, 44] trying to enhance the understanding of the tangible interaction's design space or to generalise the observations of Ullmer and Ishii. However, there is no framework or guidelines for tangible interaction that this investigation is aware off, in which the link between the type of data and the design of tangibles is addressed.

Regarding multisensory HCI, several investigations have been established to understand the nature of the relationship between these sensory channels and how to integrate them best together. Ogi and Hirose [80] investigated a multisensory environment consisting of visual, acoustic and touch sensations, combined with virtual reality HMD in order to transmit scientific data from computer to user. Their focus was on understanding the best way to enhance human's perception based on the *psychological magnitude* that each sensory channel can provide. Although, they describe the parameters used to encode data such as colour, loudness and airflow pressure. However, they have not precisely explained the reason behind this decision. Moreover, they claim that representing numerical values is possible, which is a valid claim in their case, as they used two data values between 0 and 100 at intervals of 25. However, that cannot be generalised as representing continuous values, the range is entirely different from representing discrete ones. Their studies can be considered valuable to the psychological aspect of integrating more than one sense, which is consistent with their research objective and the evaluation used in it. As they have used the magnitude estimation, which is a psychophysical method in which participants judge and assign numerical estimates to the perceived strength of a stimulus. Similarly, Harding et al. [35] constructed a multisensory system to investigate geological structures derived from the Mid-Atlantic Ridge. As such data has several dimensions, they aimed to get a better insight into the broader structure of the ocean. The prototype of this multisensory environment consists of graphics for visual sense, haptic and *sonification* (i.e. auditory interface). In their results they noted a full recognition of sound through a psycho-acoustic user study. However, they have admitted that using such a system with such data is still in his early stages. Also, they have called for more research about integrating multisensory channels to represent a different kind of data. Loftin [64], who was one of the contributors to the previous work, followed up in his article by describing the limitations of visualisations as the only model to represent data. In his reasoning, he gave an example of representing multivariate data, issuing a valid question, "*How many more surfaces could we display, with different geometries and colours, before a user becomes overwhelmed with the volume of information?*". Moreover, he promotes the necessity for expanding the bandwidth of representational variables that is limited to visual ones in most cases. As a solution, he proposes the multisensory perception which rises a new question of how to deploy these interfaces. Loftin illustrates the problem by saying "*unfortunately, few researchers have examined the question of what data is best expressed in what way*". Therefore, it is evident that identifying and understanding the set of variables that can convey data, is a crucial problem in different domains. Despite its importance, there is still a lack of research towards a general solution. Nesbitt tried in several papers to model the design space of multisensory interfaces, specifically visual, haptic and auditory. In his former work [74], he tried to extend the structure of the visual design space to haptic and sonification, then correlate this space with another classification of the design space based on metaphors. However, regardless of the effort and the good work he did to some extend, his metaphorical study is not applicable in several cases, for example, one cannot extend the same capabilities of vision to the other sensory channels [15]. As well as the transformation from raw data to visualisation through the visual encoding, cannot be literally applied on auditory or haptic interfaces. For example, a 2D view is easily represented by a line in visualisation, but how to represent a line through sound? These type of interfaces need an entirely different approach to encoding data. Nesbitt himself confirmed this in his later paper by quoting the psychophysical theory, Such that "*the psychophysical theory states that each sensory modality has distinct patterns of transduction. So, each sense has unique sensory and perceptual qualities that are adept with certain kinds of complex information*" [75]. Also, some of the examples he used to demonstrate his model, are not generalised to cover all data scenarios. In his later work [75] he used specialisation, generalisation and aggregation terms from software engineering in order to make a hierarchical framework that serves as a taxonomy for the design space. Although, he did the same thing by extending concepts related to visualisations. Though, the way he divides the design space is arguably possible, as abstracting the metaphors of the mappings between data attributes and sensory properties are reasonable. Yet, the proposed categories spatial, temporal and direct metaphors are applicable for visualisation, they have to be validated for the touch and auditory senses. Also, the mapping does not consider all data types and it is not very clear how the mapping should happen. As Nesbitt clarifies that it is the designer's task to decide which attributes of the data should be mapped to each sense [75]. The number and type of possible variables that each sensation modality can provide were not addressed in both papers. Paneels et al. [82] reviewed the designs of haptic modality for data representation purposes. Their focus was to classify designs by their representation intent, such as charts, maps, signs, networks, diagrams, images, and tables. The paper provides a comprehensive reference for researchers and designers about the prevailing practices in the design of haptic data representation. Although this work was not targeting to provide insight into the variables in the haptic modality, it succeeds in achieving its objectives by providing a detailed reference for others to look for patterns within each category of design; eventually one can propose a set of possible variables based on these patterns. In this thesis, we also look for such classes based on designs from HCI as a first step towards identifying the physical variables that each modality can provide to represent data.

Several researchers illustrated the advantages of using multisensory channels to represent data [110, 78], especially the potential of expanding the dimensions of the representation [108, 40]. Others investigated the design of a specific modality, such as sonification [123, 122, 36]. However, Roberts and Walker [90] pointed out some challenges regarding the use of all our sense to represent data. They discussed that there is lack of theoretical knowledge about several aspects, such as the data enhancement for multisensory representation, the perceptual variables for each sense, the transference of methods and designs from one domain to another, the integration of interfaces together and the construction and use of multisensory devices. Robert et al. also proposed a hypothetical procedure for the process of multisensory representation based on the one from data visualisation, such that the user should prepare the data, process the information, including data wrangling and select what they wish to demonstrate, may be enhanced or summarised in some way. It then needs to be mapped to the appropriate sensory variables, this is equivalent to mapping the data to retinal variables in the visual domain that form the visualisation design. Whatever sense then displays the information to the user. This work can be considered as a complete research agenda for anyone interested in the field of multisensory data representation, as it illustrates all the problems that still need to be investigated. Also, Roberts and Walker used in their work the terms visualisation, sonification, haptification, olfaction and gustation for sight, sound, touch, smell and taste respectively. These terms can be seen as general categories that one can use to classify the underneath existing techniques in HCI.

3.2 Data physicalisation

Since data physicalisation is a relatively new field of research, there are not many investigations compared to HCI. However, some research that could be related to data physicalisation examined data sculptures as a starting point. Zhao and Moere [124] proposed data sculptures as data-based physical artefacts for artistic and functional qualities. In their work Zhao and Moere focus on exploring the embodiment in data sculptures, as well as illustrating the benefits in using such physical artefacts to represent data. Moreover, they also propose a model to use the data sculptures as physical representations of abstract data. The proposed model lays conceptually on the *signification* from semiotics theory, where they define the physical representation of the sculpture as the *signifier* and the data set is being *signified*, which eventually results in three types of relationships; symbolic, indexical and iconic. Nevertheless, it is evident from the objectives and the proposed model that the purpose is to design data sculptures that are able to engage the audience with a central abstract idea, which is also declared by authors, "*embodiment influences how data sculptures convey insights to the audience and how, in turn, the audience perceives and interprets the metaphorical presence conveyed by the data sculpture*". Similarly, some researchers also investigated data abstraction in physical artefacts, such mapping abstract data to the physical artefact for aesthetic information visualisation [70], as well as investigating the design and implementation of data sculpture in an educational context [117]. Others have investigated the use of visual and physical representations that are embedded in the physical space as *physical referents*. However, they are not of a significant difference compared to Zhao and Moere's investigation.

Hogan and Hornecker [42] surveyed different works to illustrate the design space of what they call *Multisensory Data Representation*. They categorised the collected papers in three dimensions. In the first place based on the use of modality, where they limit the design space to the use of three modalities as the maximum. Furthermore, they defined the representational intent of each work as the second dimension, where data is either for a utilitarian intent or a casual intent. Finally, they set a third dimension called human-data relation, in which they sub-classify two different things, the nature of data and the interaction mode. In the nature of data, they considered only the way data is generated, in which data is either static (i.e. pre-stored data) or dynamic (i.e. real-time data). Although, this is a suitable approach to classify the source of data, it is not always relevant to the purpose of the representation, since the stream or the source of data is a design specification that the designers should handle in their implementation. Moreover, the interaction mode categorises interactions with data into two types, either an active interaction or a passive interaction. Although this is a generalised approach, it cannot provide the designer with the needed information to represent specific data types in the right technique or technology or the use of encodings. Moreover, their definition to multisensory data representation as "*a class of data representation that has a clear intent to reveal insight by encoding data in more than one representational modality and requires at least two sensory channels to fully interpret and understand the underlying data*" is not consistent with the proposed design model, as there is no insight about the encoding at all. Moreover, the emphasis on using at least two sensory channels to interpret the data is arguable, which is also acknowledged by the authors in their statement; "*we acknowledge that the reader may object to the dimensions of our design space and/or the terminology used to define and describe them. However, we do not intend this research to be authoritative*". However, the dimensions of the proposed design space may be acceptable for some particular applications, considering that the majority of the surveyed works is artistic work, which explains obtaining such an outcome.

Other works in the field of data physicalisation, concerns either evaluating the effectiveness of representing data physically [46], or introducing a new use case for data physicalisation [19], as well as investigating the possible interactions with the physical representations [106, 107]. However, several papers and workshops discuss the challenges of making data physical, such as Alexander et al. workshop [1] or Jansen et al. [55] opportunities and challenges in data physicalisation.

3.3 Visualisation

Investigating the design space of data visualisation can be traced back to the early twentieth century with the work of Miller [68]. However, despite many advancements in data visualisation, most of the existing related works base themselves on the work of some key players like Bertin [6]. Bertin in his work

'*Semiology of Graphics: Diagrams, Networks, Maps*' focuses on the general factors that any graphical representation can be made from. In his investigation of the properties of the graphics system, he defines the retinal variables, which consist of the two dimensions of the plane, size, value, texture, colour, orientation and shape. Then he specifies by using which variables one can make which visualisation. The definition of the visual encodings and their elements visual marks and visual variables is an outstanding work that Bertin did, which formulates clear guidelines for everyone interested in making graphical data representation. This complete blueprint describes every aspect of how such encoding should be done. Starting with the analysis of the information, through the marking of invariants and components in the data, their number and length, and towards the level of organisation of the components. On the other hand, he defines the properties and utilisation of the graphics system, such that following his guidelines will always result in a suitable graphical representation to the data.

In data physicalisation, we still are far away from such detailed design guidelines like Bertin's work. However, one has to start the investigation somewhere, and then keep accumulating towards a general and complete blueprint of the design. Therefore, in this thesis, we are trying to start the investigation in the design space of data physicalisation. The next chapter will discuss an initial starting point directed towards a more significant investigation of the physical variables. This elementary step will be based on analysing the current practices in HCI and trying to find common patterns that can be considered as physical variables.

4 The Design Space Model

As described in the related work section, most of the work in terms of frameworks for tangible interaction or HCI were focused on several aspects of designing the interface itself. Moreover, those investigated in the design space of physicalisations or multisensory representations have mostly agreed on the importance of identifying and understanding the variables or parameters from which representation can be made of. However, to the best of our knowledge there is no work where the focus is on defining which type of data can be represented through the interface or defining the set of variables that a modality has to convey data with. The only comprehensive work that is directed to the address the transformation from the digital data to the visualisation is done by Bertin [6]. Such a comprehensive work is not possible to achieve in the time frame of this thesis. Therefore, in our investigation, the focus is on answering these questions, 'What are the possible variables or parameters in different modalities?', 'What are the possible relations between these variables or parameters?' and 'Which type of data can be encoded using these variables?'. We, therefore, classify papers collected from HCI based on their modality. The consideration of modalities is linked to the current technologies in HCI, where each distinct type of interaction (e.g. haptification) is considered as a modality that can convey information to the user. Moreover, most of these emerging technologies exist in different forms and support bidirectional interaction (input and output), so that we

Phase	Title	Task
1	Familiarisation	Reading and viewing the data
2	Coding	Generating succinct labels from the data
3	Searching for themes or Abstraction	Examining the codes and collated data to identify themes
4	Reviewing themes or Structuring	Checking the candidate themes against the dataset
5	Naming themes or Categorisation	Developing a detailed analysis of each theme
6	Writing up	Contextualising the analysis in relation to existing literature

Table 4.1: Thematic analysis phases

will try to consider all these aspects in our survey. However, there will be more focus on the output that these technologies can provide, because the majority of the collected work is serving that purpose. Moreover, this survey can help those interested in the multisensory interaction to understanding the perceptual variables. Finally, based on this survey, one can list a set of physical variables that can be extracted from the surveyed technologies.

In our survey, we are based on the thematic analysis [118] of Braun and Clarke and the one of Boyatzis and Richard [12] combined. Although both analyses are theoretically equivalent, Braun and Clarke use different titles for the different phases in the analysis than Boyatzis and Richard. Therefore, we will cite both works while illustrating the steps of the analysis. The first section of this chapter presents the steps of the thematic analysis, as well as the results based on these steps. In the second section of this chapter, we present our analysis and discussion of these results.

4.1 Thematic Analysis

The thematic analysis (TA) as described by Braun and Clarke is "*a qualitative data analysis method that focuses on identifying patterned meaning across a dataset*". Therefore, the purpose of TA is to provide an answer to the research question being addressed based on the identified patterns. Patterns are identified through a rigorous process of six phases as described in Table 4.1. The flow of the phases is sequential, and each builds on the previous phase. However, the analysis is a recursive process between phases [118].

Familiarisation

This phase of analysis involves repeatedly reading and re-reading the data, in order to become immersed and intimately familiar with the content [118]. This is an important step like in many qualitative methods because it gives the researcher an overview of the data, which will help them to identify suitable codes regarding the problem [12]. In our survey, we reviewed more than 100 papers, articles and books, as well as watching interviews and video demonstrations in order to be familiar with the field and to build a good insight about the problem. After that, we extracted a subset from the reviewed collection; this subset represents a closer relationship to the subject than the excluded examples.

Coding

This phase involves generating concise labels (codes) that identify important features of the data, which might be relevant to answering the research question [118]. Usually, this phase happens in parallel to the familiarisation phase, as the researcher has to annotate the examples with common keywords and phrases that describe the significance of that paper (i.e. key characteristics of each work). This phase involves coding the entire dataset and then collating all the codes and all relevant aspects extracted from the data, together for later phases of analysis. In this phase, we have indeed recorded keywords and notes that are important for both coding the data and extracting the themes in subsequent stages of the analysis. Based on these notes, we derived an initial set of codes from the data, as depicted in the third column of Table 4.2. We followed the *inductive coding* approach, where "*coding and theme development are directed by the content of the data*" [118].

Searching for Themes

This phase of analysis is based on the extracted codes, which involves examining the codes and collated notes to "*identify significant broader patterns of meaning*" [118] that capture important details within the data (i.e. potential themes). After that, it involves collating data relevant to each candidate theme, so that one "*can work with the data and review the viability of each candidate theme*" [118]. According to Boyatzis and Richard [12], this phase is an abstraction stage, where all the codes are being abstracted in a general meaning. At this stage of the analysis, we focused on the origin of each code to extract the themes. Thus themes represent a higher level than the codes themselves and need a logical explanation for the reason for this representation, which we will do in the discussion section.

Reviewing Themes

In this phase, themes are refined, which involves combining some of them into a general theme, whereas splitting others into more specific and clear themes. The structuring process happens by checking the candidate themes against the dataset, to determine that they form a logical pattern with the data, and eventually suitable to answer the research question. Therefore, the proposed themes are discussed and reviewed by another researcher in order to identify the final set of themes. The second column of Table 4.2 represent the final set of themes.

Defining and Naming Themes

According to Braun and Clarke, this phase "*involves developing a detailed analysis of each theme, working out the scope and focus of each theme, determining the story of each*". Also, it involves deciding on an informative name for each theme; therefore it is considered as a categorisation process. In our survey, we propose the naming of each category based on the modality that each category represents. These namings (i.e. haptification, sonification, olfaction, gustation and visualisation) are popular among those working in the field of HCI and data physicalisation. Also, some researchers promote these terms to refer to a category within HCI, such as Roberts and Walker [90]. At the end of this phase, the analysis is complete, such that the codes form themes and the themes form categories. Thus all these aspects represent a coherent classification model as shown in Table 4.2.

Writing Up

Braun and Clarke define this phase as the final phase, "*which involves weaving together the analytic narrative and data extracts, and contextualising the analysis in relation to existing literature*" [118]. At this stage, we have initiated a detailed discussion section to illustrate the different aspects of the proposed model. In our analysis, we propose a more convenient naming for the categories, themes and codes, as modalities, parameters and forms respectively. Despite that some researchers called for identifying the variables in each modality/category, we have noticed that forcing the variables/parameters to be in one category is limiting the design space. As the representation can be a combination of these, which we will discuss in the next section.

Category	Theme	Codes
Haptification	Haptic-effect, weight, texture, temperature, shape, size, etc.	Tangible, force, tactile, haptic, air pressure, pressure, etc.
Sonification	Effect, loudness, direction, rhythm, duration	Speakers, instrumental equipment, auditory display, sonification systems
Olfaction	Scent-type, scent-propagation, scent-intensity, duration	Olfactory display, smell interface, ambient odor
Gustation	Taste-type, temperature	Taste interfaces, gustatory display, electrical and thermal stimulation
Visualisation	Shape, size, texture, orientation, colour, value, organisation, plane, opacity, depth	Sculptures, graphics, embedded representation, holographic imaging, surface display

Table 4.2: Result of the thematic analysis

4.2 Discussion

In the previous section, we have described the steps of the thematic analysis that have been followed to approach the results. However, we have not discussed the reason for choosing these codes, neither the transformation towards such themes. In this section, we will illustrate the origin of each code, as well as the reason to encapsulate different codes under general themes. Besides, we prefer to change the names of categories, themes and codes to modalities, parameters and forms respectively. Nigay and Coutaz [76] define modality as "*the type of communication channel used to convey or acquire information. It also covers the way an idea is expressed or perceived, or the manner an action is performed*". Therefore, a modality is a more clear naming than category, also those are working in the field are more familiar with modalities than categories. On the other hand, parameters are more specific to our investigation than themes. We actually were not looking specifically for themes that generalise the codes. Instead, our goal is to investigate the possible variables in each modality. Therefore, parameters is a convenient naming as variables are a function of different combinations of parameters. Last but not least, forms is more intuitive naming in our case than codes, because codes are mostly the names that the designers gave to their imple-

Modality	Parameters	Forms
Haptification	Haptic-effect, haptic-intensity, haptic-frequency, haptic-duration, shape, size, weight, texture, structure, pressure, force-strength, force-direction, temperature	Tangible, fingertip, mid-air haptics, ultrasound, air-flow, pressure, vibrotactile, tactile, touch, stickers, pressure, tabletop-pins, Braille display, thermoelastic, wearables, VR-grasping, force, artefacts, jamming, clay
Sonification	Effect, loudness, direction, rhythm, duration	Speakers, instrumental equipment, auditory display, audification, auditory icons, auditory earcons, spearcons, sonification systems
Olfaction	Scent-type, scent-propagation, scent-intensity, duration	Olfactory display, smell interface, ambient odor
Gustation	Taste-type, temperature	Taste interfaces, gustatory display, electrical and thermal stimulation
Visualisation	Shape, size, texture, orientation, colour, value, organisation, plane, opacity, depth	Sculptures, graphics, embedded representation, holographic imaging, surface display

Table 4.3: Proposed analysis model

mentation. Therefore, they are actually forms that each modality can be materialised in, similar to polymorphism in software engineering. Table 4.3 represents the final proposed model with the new naming and a full set of extracted parameters.

4.2.1 Modalities, Parameters and Forms

Modalities are channels used to convey or acquire information. Such that they support bidirectional communication and exist as input and output. Therefore, in our survey we investigated a lot of examples, some of them serve as implementation for the input, others serve as implementation for the output. However, the majority of examples were about the output. So that in the proposed model, we do not consider the difference between those two channels. Also, the proposed parameters are perfectly fitting as output channels. However, one still can employ them as input channels. In haptification different examples have already used the presented parameters in Table 4.3 as input channels. In sonification, one can employ these parameters for input through voice recognition and in visualisation through computer vision. As discussed earlier, each modality is a subdomain in the field of HCI, which has its own subparts. Thereby, these modalities could be seen implemented in different forms, each form has its own design guidelines and purposes. In the following, we discuss each modality with its own parameters and forms and provide some examples of these forms.

Haptification

Haptification is the modality that is concerned with all the interaction aspects of the somatosensory system, such as the design and implementation of interactions tactile and haptic interfaces, tangible interfaces and much more. However, because people are often used to discover things in their own hands, most examples of interaction are directed towards the sense of touch by hand. In our survey, we observed logical patterns when designing these interactions, which can be illustrated as each pattern represents a part in the family of haptification. These parts are what we consider as forms that haptification can be materialised in and they can be stated as the following:

Tangible interactions, Fitzmaurice et al. [29] design brick sliders to illustrate graspable user interfaces. Spindler and Dachselt [102] designed tangible lenses to explore information in a tabletop environment, similar to Ullmer and Ishii's work on the metaDesk [114]. Schmitz et al. [94] designed flexible tangibles that are 3D-printed and deformation aware to enable squeezing, binding and pressure pressing on capacitive touchscreens. Their design is a clear example of using the pressure and force as parameters for input.

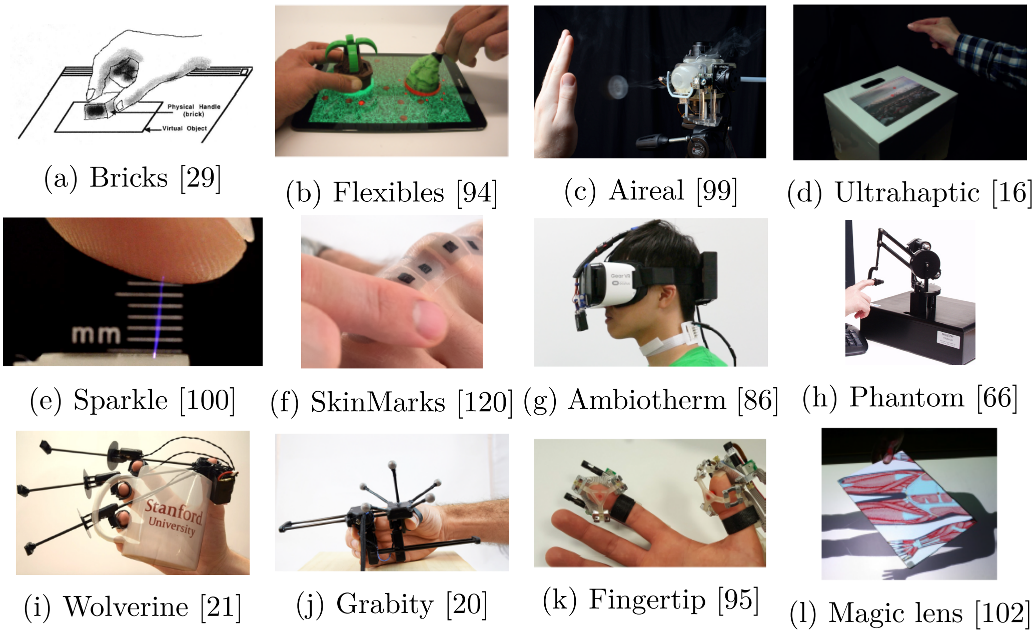
Mid-air haptics, Carter et al. [16] designed a multi-point haptic system above an interactive surface based on ultrasound. Similarly, Inoue et al. [47] used heavy airborne device to simulate an ultrasonic haptic hologram. Cha et al. [17] designed a mid-air tactile display using indirect laser radiation. Spelmezan et al. [100] designed an array-like surface that generates sparks for tactile and thermal feedback in mid-air. Sodhi et al. [99] designed a mid-air haptic device which emits a ring of air called a vortex that can impart physical forces a user can feel in free air.

Grasping in Virtual Reality (VR), Massie and Salisbury [66] designed the phantom haptic interface as the first haptic interaction in virtual reality. Schorr and Okamura [95] designed a tactile feedback for grasping in VR based on finger-mounted haptic modules. Choi et al. made two consecutive works for grasping in virtual reality, the first work [21] us a wearable interface for grasping rigid objects, whereas the second [20] is for simulating weight as well as grasping in VR. Ranasinghe et al. simulated environmental condition into VR by attaching temperature module and wind fan on the Gear VR HMD.

Some examples cannot be classified under a specific pattern, either because these are combining multiple interaction together, or because they are not enough examples to form a pattern, or because they were discussing some aspects of the haptic interaction itself. Some of these examples are the following:

Lederman and Klatzky [61] described the natural way of haptic explorations to extract object properties, such as texture, hardness or contour or weight. Besancon et al. [8] evaluated the performance of mouse, tactile and tangible input for 3D Manipulation. In their results, they noticed a similar level of precision with different completion time for the tasks. In other work, Besancon et al. [7] designed a pressure-based control for mobile by using locally-coupled pressure device. This is another example of employing the pressure as a parameter for input. Weigel et al. [120] designed innovative stickers that enable interaction on body landmarks, such that a sticker is placed on a part of the body and the user can perform different touch actions with. Stanley et al. [103] designed an innovative deformable surface based on haptic jamming. Their design is the only shape changing surface that is not based on actuated pins. Although their investigation is still in early stages, one can hope with the advances of technology to improve such a design as it can be very promising for data physicalisation as well as for simulating Ishii et al.'s vision of radical atoms. Strohmeier and Hornbaek [104] designed an actuated haptic sliders to generate different haptic textures. However, this prototype could not fully simulate textures, but it expressed a new way of using friction and force to generate different haptic experiences. Finally, Tewell et al. [111] used heat modules to generate temperature changes that are used as navigational cues for a maze exploration task. Figure 4.1 represents some of the examples in the haptification modality that have been discussed.

From all the previously mentioned forms of haptification, together with watching some video demonstrations, we have extracted a set of parameters that the designers tend to use in their implementation. These parameters are the actual channels that are used to convey data or to enable the interaction. Therefore, they can define a bigger picture towards identifying the variables. However, it is important to mention that we are not considering these parameters as final variables. Instead, we are defining these parameters



(a) Bricks [29] (b) Flexibles [94] (c) Aireal [99] (d) Ultrahaptic [16]

(e) Sparkle [100] (f) SkinMarks [120] (g) Ambiotherm [86] (h) Phantom [66]

(i) Wolverine [21] (j) Grabity [20] (k) Fingertip [95] (l) Magic lens [102]

Figure 4.1: Haptification examples

as channels that can be used individually or combined to generate a variable. Therefore, a variable is a function of a parameter or a set of parameters. This consideration will be discussed later when we cover multimodal systems and provide some algebraic demonstrations. The set of parameters extracted from the haptification forms can be stated as follows:

Haptic, The haptic parameters are haptic-effect which is the waveform or the type of haptic, haptic-intensity which is the power of the effect, haptic-frequency which is the frequency or the number of repetitions, but not the frequency of the effect, and finally the haptic-duration which is the time-interval of triggering a haptic effect. These four parameters can be seen materialised through touchscreens, vibration, electric arcs, air vortices and ultrasonic signals.

Tangibles, Through the examples of tangibles, one could extract four parameters. These parameters are shape, size, texture and structure, which were mostly embedded in the design of the tangibles. However, it is important to mention that texture and structure may represent different sub-parameters such as smoothness, roughness, softness, hardness, sponginess, rigidity, viscosity, indentation, compliance and sleepiness. Although all these aforementioned sub-parameters can be considered as full independent parameters, there is no clear design or implementation that is employing these parameters as independent channels to convey information. We also believe that these sub-parameters can be a combination of different aspects, Jansen et al. [55] gave an example that"*a smooth surface like glass or metal tends to be highly reflective, feels cold when touched, and silent when caressed. All these cues together participate in the perception of a single integrated physical property (smoothness).*

Force, From different examples, especially those of grasping in VR, one could extract four parameters. These parameters are pressure, force-direction, force-strength and weight. Finally, temperature as a parameter is seen in different examples in all forms.

The proposed set of parameters is not final and dose not cover all aspects that the somatosensory system is able to perceive. However, this set of parameters has been materialised by different developer so that there is an empirical evident about the possibility of using them as channels to convey data. Therefore, they can be helpful to investigate the design space of data physicalisation, as well as multisensory interaction. However, more empirical studies are needed to extend the set of possible parameters. Also, the investigations should be combined with psychological, physiological and neural aspects to understand the ability of users to understanding these parameters, as well as the magnitude of each parameter before being overwhelming to users.

Sonification

Sonification is usually defined as the use of non-speech audio to convey information, eventually to make data perceivable. However, in HCI, sonification is the modality that is concerned with all the interaction aspects of the Hearing or auditory perception. Besides HCI, there is the auditory displays community, which is the International Community for Auditory Display (ICAD) [1]. Auditory displays is a general name for Audification, Auditory icons, Auditory Earcons and Spearcons. Audification is defined as the direct translation of the data waveform into sound. Auditory icons are the metaphors that are considered as equivalent to icons in the desktop metaphor. An example of the auditory icon is the door-closing tone [2], which is represented by a soundwave of a door being closed. Auditory earcons are defined as the non-verbal audio messages used in the user's computer interface to provide information to the user about some computer object, operation, or interaction. Brewster et al. [14] define these earcons as "*abstract, synthetic tones that can be used in structured combinations to create auditory messages*". Spearcons are a combination of speech and earcons, as defined by Walker et al. [119] "*in that*

[1] `http://www.icad.org`

[2] `http://bit.ly/DoorClosingWave`

they consist of speeding up a spoken phrase (very recognisable) until it is not recognised as speech (more like an icon)". However, Hermann and Ritter define sonification as "*the technique of rendering sound in response to data and interactions*". Therefore, sonification is an explicit naming for the modality rather than auditory displays, as auditory displays are more about design specifications and examples from the general scope sonification.

In our survey, besides investigating the previously mentioned forms of auditory displays, which are eventually sonification's examples. We also investigated several papers of the design and implementation of sonification systems. Kaper and Weibel [58] investigated the use of sound in virtual reality to enhance the experience and to explore and analyse different aspects of scientific data. They have implemented a digital instrument for additive sound synthesis, as well as a program to visualise sound effects in a cave virtual reality environment. Kaper and Weibel in their system encoded nine different sounds that differ in their tone, frequency and amplitude. The idea was to generate a kind of a spatial effect within the different sounds, which they have achieved by recruiting two kinds of waves, one is the original sound and the second is partial that serves as a spatial enhancer. Madhyastha and Reed [65] investigated combining sonifications and visualisations. Madhyastha and Reed illustrate the reason for this combination between auditory and vision *that sound alone often cannot convey accurate information without a visual context. For example, merging the sound of a hammer with sounds commonly heard in a gymnasium, such as cheering or a referee's whistle, makes it harder to distinguish the hammer's sound from that of a bouncing basketball* [65]. Therefore, their system maps data to sound while visualisation the data, such that the sonification is used as a complementary channel to enhance the understanding through vision. However, they have also investigated using sonification as the only channel to convey the information. They noticed that in an ideal or not nosy environment, sonification could deliver a clear insight about the item being presented through sound. Hermann and Ritter [38] designed a sonification model for data analysis through an interactive tool. In their model, sound occurs if an interaction happens with the data, such as examining, exciting or playing with the data. Distinct sound effects are linked to what Hermann and Ritter define as *material design*, which is determined by the setup of the elements and the interaction between them. Such that the data more or less translated to a specific instrumental sound through virtual physics that permit a vibrational process analogous as in real sounding materials [38]. Hermann et al. [37] proceed with another work of sonification, such that they combine sonification with a tangible desk. Different sound effects are coupled with tangible cubes. The relations between these tangibles reveal different insight about the data, like most of the other tangible desks. Zhao et al. [123] developed auditory information seeking platform, which is a similar principle to information seeking on desktop or visualisation in general, such that it support gist, navigating, filtering and grouping. The platform was directed to visually-impaired users, and the sonification conveys all the information through spatial sounding. The gist is represented asa short auditory message presenting the overall trend or pattern of a data collection [123], different gist are played while navigation, finally the details are given through speech. Zhao et al. [122] recruited the platform to explore geo-referenced data. Similarly, Halim et al [34] and Grond et al. [33] developed sonification for data exploration tasks, whereas Blanco et al. [9] developed auditory assistance for monitoring. Others like de Campo [25], Grimshaw and Graner [32] and Hermann [36], investigated in the design space of sonification.

To discuss the parameter extracted from the examples, one has to start with defining the sound. The definition of a sound and its characteristics illustrate the set of parameters that we have in our table. A sound is the sensation of pressure variation in the car caused by a vibrating source [65]. Each sound has some characteristics, such as the measure of amplitude which is perceived as loudness, the waveform which determines the timbre and the frequency which is discernible as the pitch [65]. Although there are more characteristics that can be manipulated in order to generate a rich set of sounds. However, changing the frequency and waveform will result in different effects. Therefore, we prefer to name the parameter effect, such that the pitch and timbre lay under the effect. However, the amplitude or loudness is a distinct channel, and it has to be an independent parameter. As one effect can be with different amplitude and therefore it can be received differently. Also, loudness is sometimes referred to as a volume. Therefore loudness and volume are interchangeable terms. We prefer to use loudness as it was more common amongst the surveyed examples. The direction of any recognisable sound is perceived by users, such that a user can distinguish two different sound effect being generated from different directions. Also, the direction is the main component in spatial sounding. However, some prefer to call it location, others call it direction. In our investigation, the direction was more common, and therefore it is the chosen name for the parameter. Rhythm and duration may seem interchangeable. However, rhythm here represents the number of iterations that a sound effect occurs. It is precisely the frequency of playing a sound effect. However, the word frequency may be confusing if used, as one may think that this frequency is similar to the pitch or even to the waveform. Therefore, rhythm or windowing are more general, and rhythm is a common word in the context of sound, whereas windowing is more common in the context of signals. Thus, rhythm is chosen as a name to the parameter. Whereas duration refers to the time interval wherein a single sound effect is played. For example, if a sound effect had a length of two seconds and played for three times, then the duration parameter is two and the rhythm is three. Such that the period of playing this sound effect for three times is not considered.

Hermann and Ritter noted essential challenges when deploying sonification. First that *there is no unique way of mapping between components and parameters. The listener, therefore, requires some learning time to get acquainted with a chosen mapping. The necessity of parameter assignment leads furthermore to a combinatoric explosion of possibilities with increasing dimensionality* [38]. Second is about the relationship, such thatthe sonification is just the superposition of independent events. This sonification method cannot exploit the relationship between different data points, e.g. the local density of the data [38]. As well as that the dimensionality of the sonification is limited by the number of parameters. Although these challenges are true, they can be solved in different ways if the design space of sonification has been fully covered. One can, for example, come with standards for the mapping of sound effects and data point based on the possibilities of sonification. Moreover, the relationship between data points can be represented through the combination of different parameters. However, one still has to investigate the magnitude of each parameter and the number of possible combinations between these parameters, as well as the number of the sound effects in both the temporal and the spatial domain that can be presented to the user without being interfered with each other. Nevertheless, sonification still can provide a lot for data exploration and analysis task, even if not as the main modality, but at least as a complementary modality that encodes some dimensions of a data being represented through different modalities.

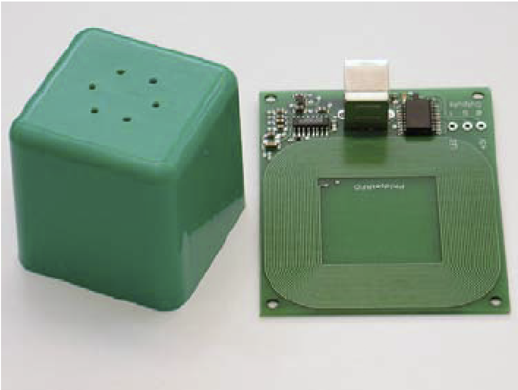
Olfaction

Olfaction is a term that is given to the sense of smell, in medicine it is defined as a set of *chemoreceptors* [3] that forms the sense of smell, which occurs when odorants reach to specific sites on olfactory receptors [4]. Similarly, in HCI, olfaction is the modality that is concerned with all interaction aspects of the olfactory system. Unfortunately, the olfaction modality as a research domain has a limited number of studies in comparison to haptification or even sonification. However, in our survey, we investigate several examples

[3] `https://en.wikipedia.org/wiki/Chemoreceptor`
[4] `https://en.wikipedia.org/wiki/Olfactory_receptor`

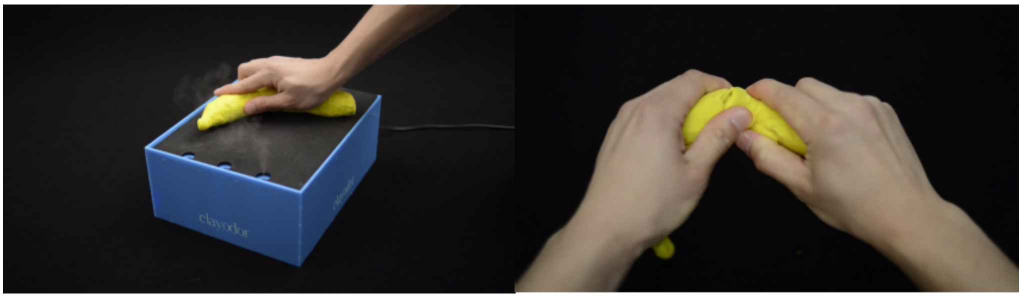
from the existing literature. Brewster et al. [14] investigated the use of smell to enhance users' memorability in comparison to the use of normal text in a searching task. For their study, they developed a simple photo browser that allows for text and smell tagging. Users then had to use a set of smell cubes with RFID tags attached to them. The user can add one or more tags to a photo by moving the cube over the RFID reader. For searching a digital photo in a collection, the user can move a cube over the reader, and the browser will retrieve all the photos with that smell. Similar operations are allowed for text. The results of the study showed that participants could tag effectively with text labels. This is reasonable as tagging with text is a common and familiar task. However, the performance with smells was lower, but the authors claim that "*participants performed significantly above chance, with some participants using smells well*" [14]. Kao et al. [57] developed a smell-generating device that changes the smell based on user manipulation of a clay-like malleable material. Different shapes are predefined in the system, and each shape is linked with a smell. When a user puts the deformed shape on top of the device, the system will detect the shape based on the pressure area of the shape and then will compose the corresponding smell of this shape. Covaci et al. [23] investigated the contribution of olfaction to users' learning performance in a serious game, where different smells are presented while the user is playing. They compared results of performance in the presence and absence of olfactory feedback, which according to the authors, showed that the performance is improved with the presence of scent and also the users had more joy in the game. Similarly, Murray et al. [72] investigated the user perceived quality of experience when multimedia is enhanced with olfaction. The results show that the scent type influences the experience of users pleasantly and unpleasantly, based on the scent type. Finally, Obrist et al. [77] investigated the experience of smell and emotions that accompanied with it, based on stories collected from users through a questionnaire. According to the result of the questionnaire, they define categories of smell experience, such as remembering through smell, changing the mood, attitude and behaviour with the smell, overwhelming power of smell, etc.

Based on the collected examples in our survey, we observed four common parameters for olfaction. The scent-type is the most effective parameter in the set, as it represents a distinctive odour. Thus it has the highest magnitude since we have a large set of scents in the world. The scent-propagation and scent-intensity are partial parameters, since scent-propagation is mostly stochastic, and scent-intensity can be perceived differently from user to another based on their olfactory system. However, scent-propagation still can encode some information in an exploratory task, for example, a user walking



(a) Olfoto [14]

(b) Meta [73]



(c) Clayodor [57]

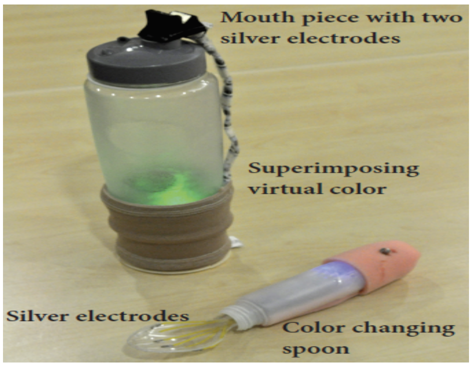
Figure 4.2: Olfaction examples

in a room can smell a scent in only one part in that room and the scent is absent in all other parts. Whereas the scent-intensity can be adjusted to the user's preference, for example, a system allows configuration to happen and the user can set several values to the intensity based on the trail of the system before use in a real task. Finally, based on the examples, the duration that a scent is present can be considered as a parameter, for example, a longer presence of a scent define a larger value [23]. However, duration as a parameter is also debatable, since some may argue that after some time the user will be saturated with a scent and will not be able to distinct time intervals of a scent. This argument is definitely valid if the same scent-type is used at different duration. However, using different scents with different intensities is theoretically distinguishable.

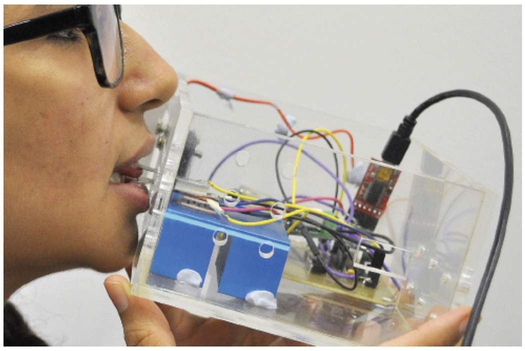
Gustation

Gustation is the sensation of taste that is produced when a substance reacts chemically with the taste receptors in the mouth. Thereby in HCI, gustation is the modality that is concerned with all interaction aspects of the taste or gustatory perception system. Similar to olfaction, gustation also has a limited number of studies, which can be explained as the gustation system is complex and different aspects are involved in the experience of taste, such as the visual appearance of the substance and the smell [73]. Also, there is still no advances in the technology related to gustation such that a device can generate different tastes dynamically. However, in our survey, we present some attempts to enable interactions through taste.

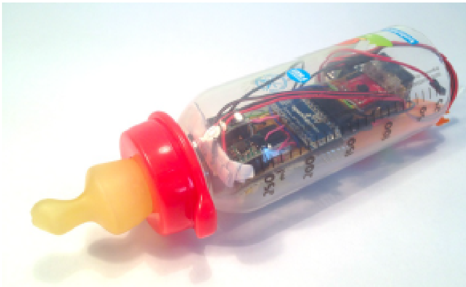
Ranasinghe et al. [87] designed an interface to share taste and smell sensations digitally over a network. The interface is made of an actuator as a control system to simulate the taste through electrical or thermal stimulations on the tongue. According to them, results suggested that sourness and saltiness are the main sensations that could be evoked through this device. Ranasinghe et al. proceed their investigation in the digital taste and smell interfaces with different works. In the first paper [88], they developed a set of utensils that uses electrodes to stimulate different taste sensations. Whereas in the second paper [89], they improved their former digital flavour interface by allowing it to produce smell using a controlled scent emitting mechanism, such that when using the device, virtual flavours are delivered as a combination of digital taste sensations and different scents. Distinct taste sensations are delivered using electrical and thermal stimulation on the tongue, while smell sensations are delivered by heating an array of solid perfumes. Murer et al. [71] investigated the potentials of taste as a playful interaction modality through an exploratory study. For their study, they designed different interactive lollipops (i.e. tangible artefacts) that serves as a controller that provides taste-based output while allowing some degree of tangible input through moving around its handle. Samshir et al. [93] developed a taste interface to create sweet sensations by manipulating the temperature on the tongue and without using chemicals substances. The temperature degree changes within a short period on the surface of the tongue in a range of 20 °C, between 20 °C and 40 °C. The authors claim that users were "*able to obtain the sweet sensation purely from the device and in another scenario, enhancement of sweet sensations was reported while tasting sucrose after using the device*". Finally, Spence [101] wrote an article about opportunities and challenges in deploying taste in multisensory interactions. Spence discussed how different senses contributes to the perception of flavours and the need for studies over the multisensory perception of flavours. Spence illustrates that the same rules of multisensory integration that have been thoroughly explored in interactions between audition, vision, and touch may also explain the combination of the (admittedly harder to study) flavour senses.



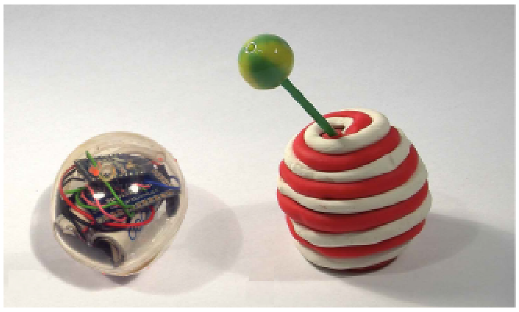
(a) Taste+ [88]



(b) Digital Flavor [89]



(c) Baby Bottle [71]



(d) LOLLio [71]

Figure 4.3: Gustation examples

From the surveyed examples, one could notice two common patterns when designing interfaces for the gustatory perception. Mostly, two parameters were being handled; those are the taste (i.e. flavour) and the temperature. The taste concerns all aspects related to the experience of flavours, such as sweetness, saltiness, bitterness, sourness and umami. However, all these forms lay under the general scope of taste, and therefore we prefer to call the parameter taste-type. On the other hand, temperature differences on the tongue can lead to a different experience of taste. Although this experience can differ from a user to another user, however, the temperature as a parameter still can convey unique information. Finally, some examples showed the effect of using flavours repeatedly at different time intervals. However, the duration and frequency still need to be investigated to be sure that they can convey information. Therefore, we do not consider them as independent parameters at the moment, as we believe that users can get saturated with the flavours after some time.

Visualisation

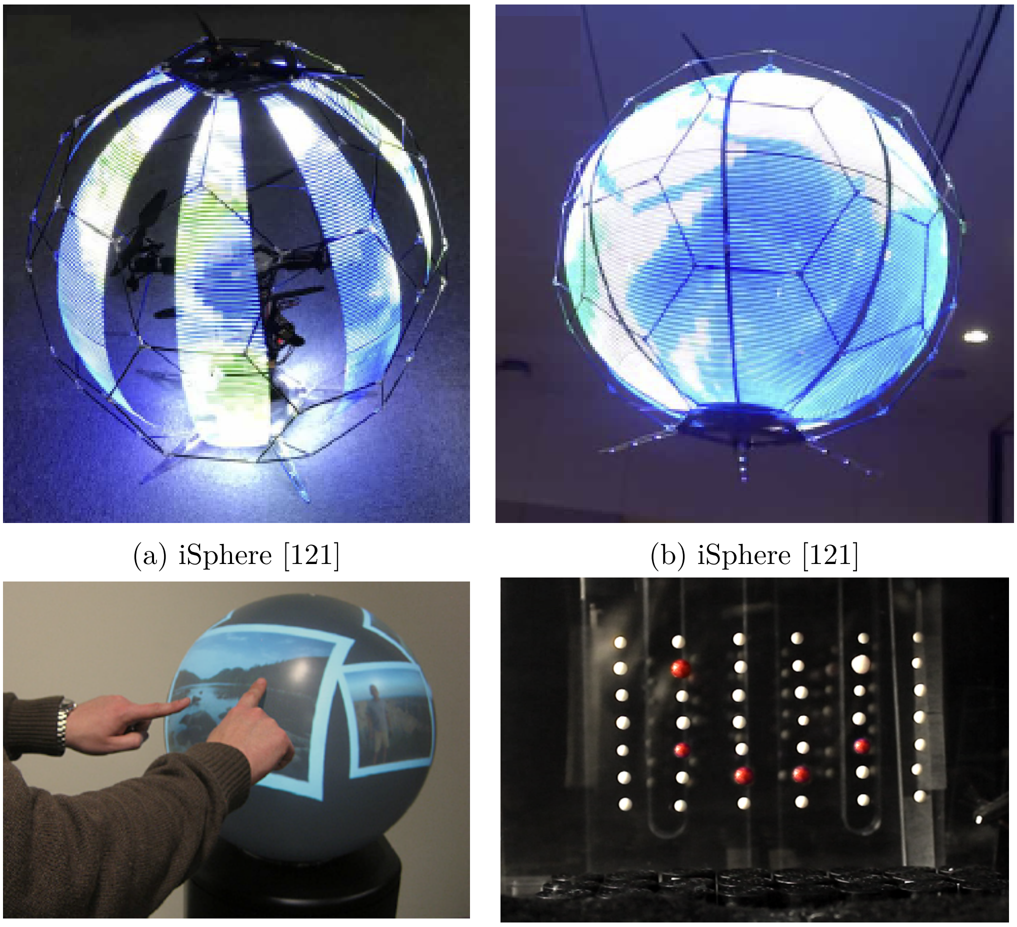
Visualisation concerns the applications that are related to the visual perception. The visual sense of sight is arguably considered as the dominant channel to represent information [68], and this consideration can be linked to the ability of the eye to process in a spatial and atemporal way as explained by Bertin [6]. The use of visualisation can be traced back to the ancient time where visualisations were used for communication such as the cave paintings and hieroglyphics writings, which are considered as symbolic visualisations. Similarly, the research in the field of visualisations has existed for a long time, and there is a lot of studies and applications for visualisation. However, in this survey, we address only several examples since the ultimate goal of this thesis is not to cover the whole visualisation.

Regarding graphics, we consider the work of Bertin [6] as our primary reference. Although there are many studies established after Bertin's work, Bertan's work remains comprehensive on the aspects that concern graphics, diagrams, networks and maps. Also, the retinal variables that Bertin proposed are all considered as parameters in our classification. On the other hand, there are a lot of applications and methods for visualisations that are beyond the static graphics. Moere [69] discussed the design of he calls persuasive ambient visualisation. In this discussion, he presented different applications for ambient information visualisation, as well as providing several examples of the recent evolution of ambient displays, from merely aiming to inform people about data patterns towards increasing awareness of themes underlying the data. Pousman and Stasko [85] investigated in the same matter and came up with a taxonomy of four dimensions for these ambient displays, mainly classifying the examples based on their application. Sahoo et al. [92] developed an interactive mid-air display using acoustic levitation. They used objects that have different faces which are coated with different materials to present different physical properties, eventually to make an interactive physical virtualisation. In our survey, we also noticed that there is an increasing interest in the spherical displays. Benko et al. [5] developed a multitouch spherical display that uses an infrared camera for touch sensing and a projector for display. Crespel et al. [24] developed a framework for developing applications for multitouch spherical displays that makes it possible to create interactive content by programming standard GUI applications, such as interactive web pages. Yamada et al. [121] designed a flying spherical display using a drone and multi-LED tapes around the drone, in order to physically and directly emerge arbitrary bodies in the real world. Finally, the last example that we addressed for visualisation is the one of Dalenius et al. [112], which they have investigated holographic data visualisation using synthetic holography to share information.

As previously mentioned, the set of the proposed parameters lays largely on the retinal variables that are forwarded by Bertin [6]. The variables, shape, size, texture, orientation, colour, value and two axes of the plane that Bertin has set out, together with their level of organisation are included in our classification. However, the value as a variable can be confusing as all the variables have value, which is also the reason that Bertin [6] put it there. Bertin illustrates that the value can be interchangeable with intensity, which can be considered in different ways, such as measuring the quantity of reflected light (i.e. brightness) or measuring the ratio (i.e. contrast). Besides the retinal variables of graphics, we address two other parameters; those are depth and opacity. The depth of an object in the 3D space is uniquely perceivable, such that two identical objects at different depths are perceived differently. On the other hand, the opacity of objects refers to their transparency. Therefore it is not similar to the texture where objects are organised in a specific matter, neither included into colour where only the gradient varies. Therefore, the opacity is an independent parameter, and it can be essential if a transformation happens from visual to physical representations. Finally, the proposed set is not final, and other parameters may be included, such as the motion. However, based on the examined examples, there was clear evidence about the parameters included in Table [**?**].

4.2.2 Systems

After discussing the different forms and parameters of each modality, it is important to discuss the examples that do not belong to one modality. Instead, many examples are developed in a way that they form a system that could be seen as a combination of different modalities. Some of these examples that have been addressed in our survey are the implementation of shape-changing displays, which could be seen as a combination of haptification and visualisation. Ishii et al. [49], designed a shape-changing table to simulate the embodiment of their visionary radical atoms. Similarly, Siu et al [98]. designed a mobile shape changing device for tangible interaction and grasping in VR. Both examples are similar to Follmer et al. [30] example, which designed a shape-changing display for dynamic physical interactions. However, Leithinger and Ishii [63] were the first to investigate the shape displays in their work Relief. Leithinger et al. [62] followed with another work called sublimate by integrating actuated shape displays and augmented reality to demonstrate interactions with shape displays. Benko et al. [4] from Microsoft research team developed a handheld shape rendering controllers to simulate touch, shape and texture in VR. Benko et al.'s work can be considered as



(a) iSphere [121]
(b) iSphere [121]
(c) Sphere [5]
(d) Joled [92]

Figure 4.4: Visualisation examples

the smallest prototype of a shape-changing interface and they also relatively succeed to simulate different textures and shapes.

Other examples that we have addressed are a combination of different modalities, such as sonification, visualisation, etc. Hollerer et al. [43] designed a large-scale instrument for immersive visualisation and simulation. Their design which they call Allosphere, consist of high cubical space comprises an anechoic chamber with a spherical display screen of ten meters diameter, surrounded by a bridge structure that allows up to thirty users to stand on. The Allosphere is a virtual reality environment with a focus on the size and providing collaborative experiences. Besides the high-resolution active stereo projectors, the system is equipped with a complete 3D sound system with hundreds of speakers. Such a system employs different modalities and provides different interactions, such as combining visualisation, sonification and a sort of haptification through direct touch. Milczynski et al. [67] developed a malleable interaction surface that looks like a tangible tabletop interface, combined with sonification, where data are acoustically explored through an informative interaction of sound. Narumi et al. [73] developed a system that combines gustation, olfaction and visualisation through VR. They hypothesise that the complexity of gustatory sensation can be applied to the realisation of a (pseudo-gustatory) display that presents the desired flavours through a cross-modal effect. The goal of their design was to investigate the effect of other sensations on taste, such that they tried to change the perceived taste of food by changing its appearance and scent. In their exploratory study, they claim that the system can change the perceived taste and that this enables the user to experience various tastes simply by changing visual and olfactory information without changing the chemical ingredients in the cookie [73].

All the previously mentioned examples cannot be classified under a specific modality; instead they are a result of a mix of modalities. Therefore, they should be handled differently, and one has to understand the set of parameters at the scope of systems instead of the scope of individual modalities. Although, this consideration may increase the complexity of the research. However, we believe that it is the rational way in order to understand the design space of data physicalisation, as well as one of the multisensory interactions, such that one may be able to answer the question of the perceptual variables [64] in multisensory interactions or physical variables [55] in data physicalisations. Other researchers also support our belief, such as Loftin [64] and Jansen et al. [55]. Jansen et al. [55] illustrate that "*considering senses separately does not fully capture the way information is encoded and accessed in the physical world*". We, therefore, provide our model of analysis in term of systems. A system is a general scope that can be made of one or more modalities, such that a system can be a physicalisation or any multimodal interface for data representation. Thereby, a system can be represented mathematically as illustrated in equation 4.1, which can be explained as the following:

- A dataset D that has x dimensions represented as $D = [d_1, ..., d_\mathrm{n}, ..., d_\mathrm{x}]$, needs to be represented in a system.
- The parameter g_k from the domain of parameters $G = [g_1, ..., g_\mathrm{k}, ..., g_\mathrm{y}]$, is a function over the data point d_n. Thus, parameters are at most 1-dimension and they can only take one data point. Eventually, the set G can represent up to y data points.
- A variable f_j from the domain of variables $F = [f_1, ..., f_\mathrm{j}, ..., f_\mathrm{z}]$, is

created from the mapping of one or more parameters, such that the variable f_{j} is a function over parameters. Eventually, a subset of parameters can create one or multiple variables.

- Finally, a system s_{i} from the domain of systems $S = [s_1, ..., s_{\mathrm{i}}, ..., s_{\mathrm{h}}]$,is constructed by applying certain operations over different variables, such that s_{i} is a function over variables.

$$
\begin{aligned}
& g_{\mathrm{k}}(d_{\mathrm{n}}) \\
& f_{\mathrm{j}}(g_1, g_2, ..., g_{\mathrm{m}}) \\
& s_{\mathrm{i}}(f_1, f_2, ..., f_{\mathrm{r}}) \\
& \text{Where:} \\
& d_{\mathrm{n}} : \text{is the } n^{\mathrm{th}} \text{ data point from a dataset} \\
& g_{\mathrm{k}} : \text{is a function of mapping a parameter to a data point} \\
& f_{\mathrm{j}} : \text{is a function of mapping set of parameters to a variable} \\
& s_{\mathrm{i}} : \text{is a function of mapping set of variables to a system}
\end{aligned}
\tag{4.1}
$$

By representing physicalisation in mathematics and applying certain algebra, we can explore models that will provide a better understanding of the design space physicalisation. For example, to answer the question of how can one convert a classic visualisation to physicalisation, a preliminary observation can show that parameters that are shared between two variables, can be directly converted from one form to another. To elaborate more over this example, let us consider the following scenario: A visualisation system has a variable A, and a haptification system has a variable B like depicted in figure 4.5, the parts that intersect, mean that they have the same parameters, such that variables can have intersection by parameters. From that preselective, one can have one to one direct mapping when they try to convert from visualisation to physicalisation. For example, in shape-changing displays, the parameter (shape) is shared between the visual appearances of the pins and the tangible perception when one touches the pins. Another example if one had a 3D model in a computer, the direct mapping of the parameters (shape and colour) can let to transform this 3D visualisation model to a physicalisation form, through a simple 3D printing. However, for the parts that they do not intersect, one cannot just do a direct one to one mapping. For example, if one had a visualisation of line, then it is not clear how to map that to haptification. One naive way of representing a line through haptic can be by using different haptic motors placed in a line. However, the user might not understand the resulting haptic physicalisation due to the temporal nature of haptic [6], such that the user has to move his hand over the line, which

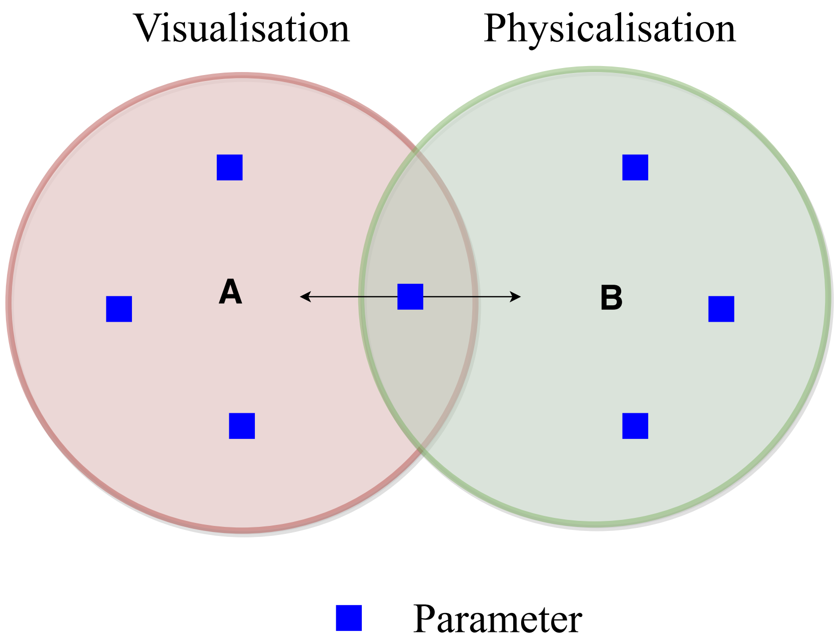
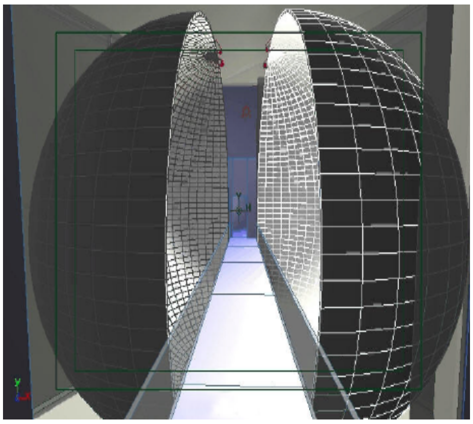


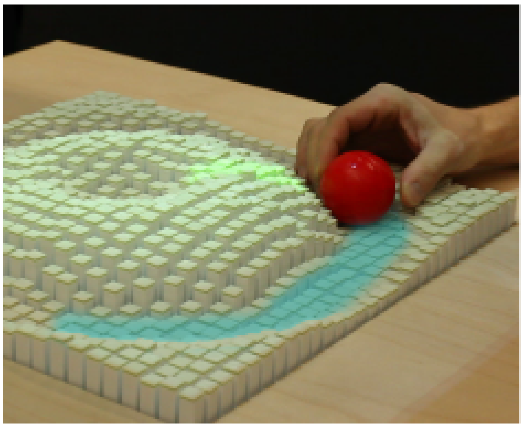
Figure 4.5: Parameters intersection

can lead to misunderstanding. Even though a single line might be understood correctly by the user, there are still examples such as a strip-line or a curvature, which will not be readily understood by the user through a simple haptic interaction.

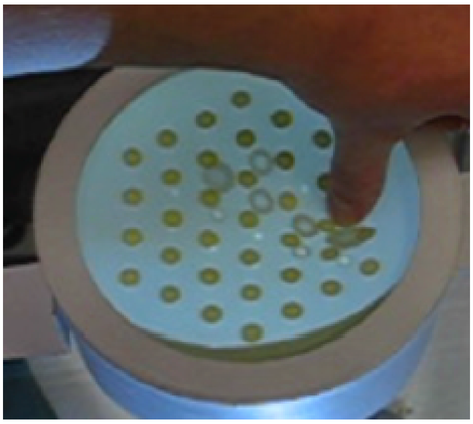
Of course, at this stage, this analysis is rather crude, and there is a need for more in a detail investigation on how to create such sophisticated models analysis. Therefore, finding a mathematical representation of the design space of data physicalisation can facilitate the task for designers when designing physicalisations, since designers can relate to these mathematical representations in their design.



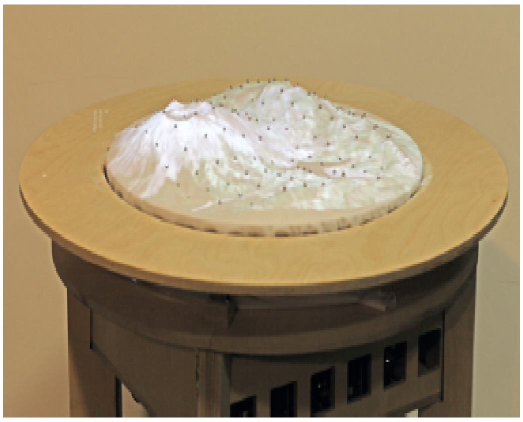
(a) Allosphere [43]



(b) inForm [30]

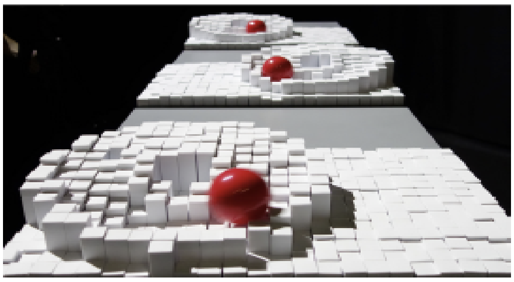


(c) Malleable Sonification [67]

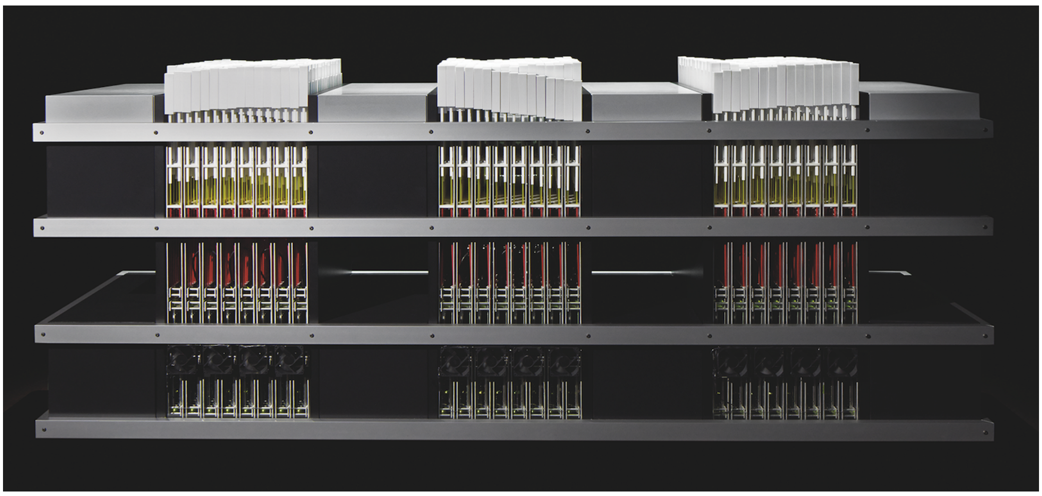


(d) Relief [63]

(e) Shape Shift [98]



(f) Transform [49]



(g) Transform [49]

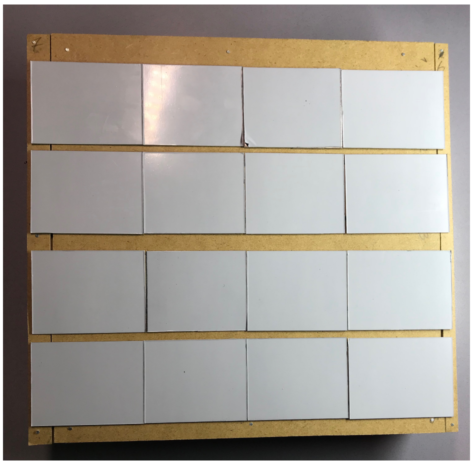
Figure 4.6: System examples

5 Implementation

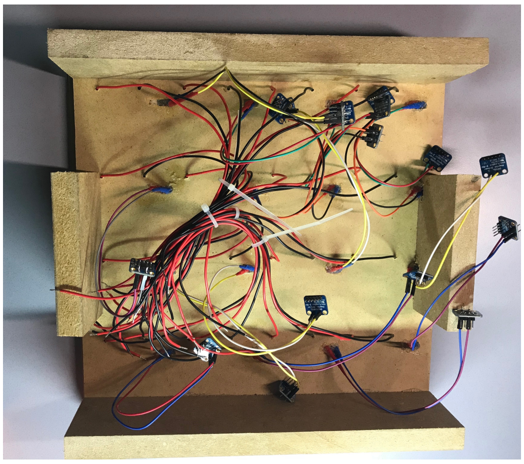
In this chapter, we discuss the design and implementation of a *thermo-tactile* interface. This prototype serves as a proof of concept in harnessing different modalities or physical variables as described in the previous chapter to enable physical representations. Our prototype can be used in different experiments to

- Evaluate the usability of physical variables in representing different types of data.
- Evaluate the usability when using different variables as parallel dimensions when representing multidimensional data.
- Observe the limitations of the selected variables with respect to the current technology.

The main idea of the prescribed thermo-tactile interface is to build a system that can represent data individually as a stand-alone platform and also as a complementary platform to extend the traditional ways of representing data, mainly visualisations. The thermo-tactile interface is basically a surface-like system. The name thermo-tactile is given to the prototype based on the two non-visual modalities that it supports, namely heat and haptic (thermal and tactile). The data is represented in the form of feedback being given to the user through different components, which constitutes the surface. The full



(a) Front View



(b) Back view

Figure 5.1: The thermo-tactile prototype

configuration of the thermo-tactile interface as depicted in figure 5.1, consists of sixteen peltier element, as well as sixteen vibration motors, which both forms an array-like surface. The components that construct the prototype are discussed in details in the following three sections; the hardware, software and interface design. The hardware section describes the components of the thermal and tactile sub-parts. Also, the hardware section describes the theoretical concepts behind each component, such as describing the thermoelectric modules and the electronic circuit that is controlling them in terms of the design and schematics. Similarly, we can describe the DC vibration motors and the haptic drivers to control them together with the full circuit description. On the other hand, the software section covers the programming steps to run the components as one coherent system. In this section, the control logic, the inter-component communication, the external communication with a computer and the hand tracking are discussed in detail. Finally, the interface design section illustrates the design pattern behind the thermo-tactile system, the surface design and assembling the components in one system.

5.1 Hardware

The thermo-tactile system consists of two parts. Each of them has its electronic circuit and logic. Nevertheless, each part has also an individual microcontroller that processes a different type of data and has an independent logic. In other words, they are fully isolated as two distinct components.

Moreover, the isolation also applies to the power sources. Since each part has a different power source, these are 12V and 5V to the thermal and the haptic respectively. Although it was possible to aggregate the two parts on the same printed circuit board (PCB) without external hardware, this applies for the main components, but the power source for example still needed an extra voltage regulation circuit in case of using the 12V as the main power source for the haptic subsystem. However, the preference was to follow best practices and to comply to the separation of concerns pattern when designing a system of systems [3]. The reason for that separation is to facilitate the maintenance in case of failure. Also, to give the flexibility to reform, alter and reconfigure the system in the future. The thermo-tactile in its current form allows for reconfigurability, such that each part can form an individual interface on its own. Therefore, two surfaces can, for example, be remade from the current surface. Moreover, the current design is flexible to extensions, such that other modalities can be added to the system in the future. Finally, considering the aforementioned flexibility in the current design and as the technology advances in the future, such that smaller thermal and haptic module made available, one can reconstruct the system into an artefact that synthesises an extension to the TangHo [97] platform in a similar way of Snake Charmer [2].

5.1.1 Thermal Interface

The thermal part is made of sixteen thermoelectric modules. Each Peltier module is joined with an electronic circuit that functions as a switch to the Peltier module *Peltier* effect 5.1.1. However, the electronic circuit is not just about a simple switch, but, it also works as a voltage regulator to control the amount of voltage passing to the thermoelectric module. Thereby, we can tune the heat value since it is directly proportional to the voltage. The electronic circuit to control a single Peltier illustrated in 5.2 consists of an Arduino Mega [1] as a microcontroller, a one kilo Ohm resistor, an N-channel MOSFET, the Peltier module and a 12V/100A DC power source. The following subsections we descibe each element of this electronic circuit.

Microcontroller

Arduino [2] is a popular opensource hardware and software platform mainly used for educational purposes, IoT and small to medium scale embedded systems. The Arduino ecosystem provides different options for microcontrollers,

[1]`https://www.arduino.cc/en/Main/ArduinoBoardMega2560`

[2]`https://www.arduino.cc`

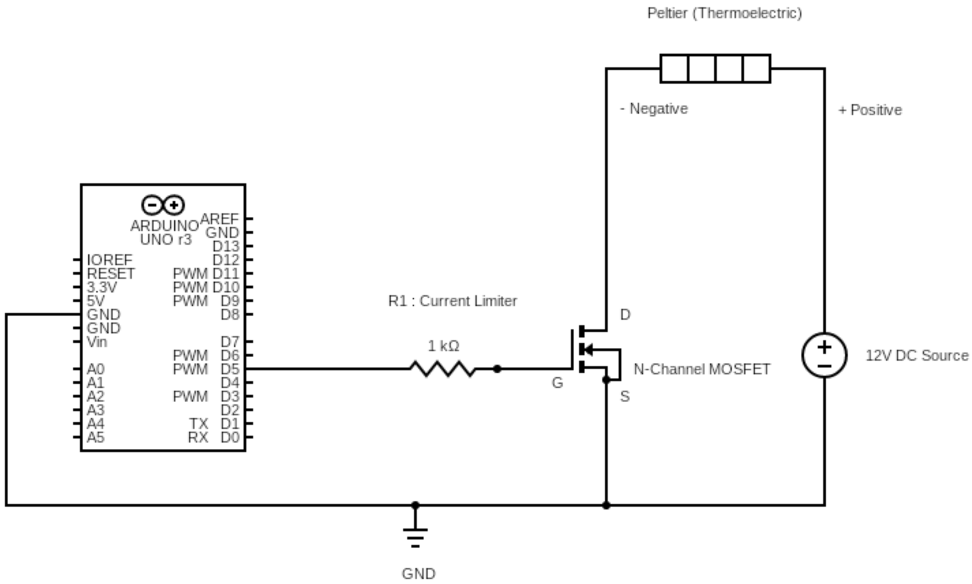


Figure 5.2: Single Peltier control circuit diagram

such as UNO, NANO and MEGA. Furthermore, it is relatively straightforward, easy to use and has an active community. From this point of view and because of some of the other components have already existing libraries that are compatible with Arduino, choosing Arduino as a platform for the controllers was straightforward. Moreover, the decision to use an Arduino MEGA as the main microcontroller for the thermo-tactile system is because it is based on the Atmel Atmega 8-bit microcontroller, specifically ATmega2560 [3], while the other two aforementioned based on on the ATmega328 and ATmega328P respectively. Although these three microcontrollers have the same clocking speed at 16MHz, they differ in different means, such as the flash memory, SRAM, general purpose input-output pins (GPIO), analog pins and most importantly the number of pins that can provide pulse width modulated signals (PWM). The Arduino MEGA outscores the UNO and the NANO in all of the previously mentioned specifications. Figure 5.3 shows the mapping of ATmega2560 ports to Arduino Mega pins. The MEGA2560 can handle more complex projects, since it has 54 digital I/O pins, 16 analog inputs and more space for sketches as it has 256 KB of flash memory where the code is stored. Moreover, the Mega board like most of the Arduino boards that comes programmed with a default bootloader, and one can directly up-

[3] `http://bit.ly/Atmega2560`

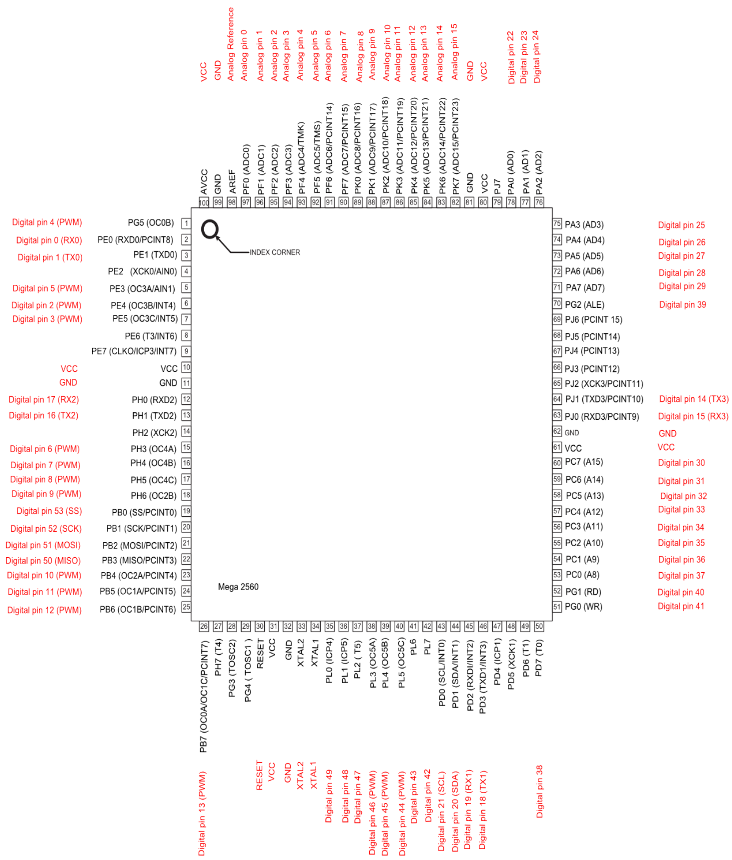


Figure 5.3: Mapping Between arduino mega pins and ATmega2560 ports

load new code to the board without any third-party software. However, it is necessary to mention that the burned bootloader uses the STK500 protocol [4] for communication. It is sometimes necessary to handle this manually if one decided to upload the board from outside the default software development kit that is provided by Arduino, which was the case while implementing the software of the thermo-tactile system.

MOSFET and Current Limiting Resistor

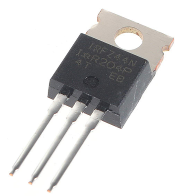
The metal-oxide-semiconductor-field-effect transistor (MOSFET) is a very popular type of field-effect transistor (FET), which is most commonly fabricated by the controlled oxidation of silicon [13]. Each MOSFET 5.4b usually has three external leads, formed by the Gate(G), Drain (D) and the Source (S). The gate is insulated and its voltage determines the conductivity of the device, more specifically the conductivity of the channel between the drain and the source. This ability to change conductivity with the amount of applied voltage can be used for amplifying or switching electronic signals. Moreover, the main advantage of a MOSFET is that it requires almost no input current to control the load current when compared with bipolar transistors [13]. When the channel of the MOSFET is composed of a majority of electrons as current carriers, it is called N-Channel MOSFET. When the MOSFET is activated and is on, the majority of the current flowing are electrons moving through the channel. The other type of MOSFET is the P-Channel MOSFET, where the majority of current carriers are holes. However, for the implementation, it is more important to discuss the operation of the MOSFET, especially N-type MOSFET for switching, rather than diving into the theoretical concepts behind the MOSFET and its fabrication. The MOSFET generally operates in two modes, enhancement mode and depletion mode. A depletion-type MOSFET is normally on. In other words, a drain-source channel exists without a voltage on the gate. In this case, maximum current flows from drain to source and there is no difference in voltage between the gate and source terminals. However, if a voltage is applied to the gate lead, the channel between the drain and the source becomes more resistive. Thereby, less current flows until the gate voltage is so high, that the transistor completely turns off. An enhancement mode MOSFET works the opposite. It is normally off when the gate-source voltage (VGS) is zero. However, if a voltage applied to the gate terminal, the conductivity of the device increases, which means the drain-source channel becomes less resistive, subsequently more current will flow in the channel. Thus, achieving the

[4] `http://bit.ly/STK500_Protocol`

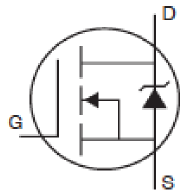
switching mechanism to control high-current using MOSFET is simple, for example by employing the enhancement as in the thermal control circuit 5.2. A 12V DC power source is connected to one terminal of the load, which is the Peltier. The second terminal of the Peltier is connected to the drain. The source is then connected to ground, and the gate is connected to the output of the microcontroller while the microcontroller is also earthed to the common ground. When the output pin of the microcontroller is low (zero), the channel does not exist and no current flows in the circuit. The opposite happens when the output pin of the microcontroller is high, the difference of voltage between the gate and source allows current to flow through the load, through the drain to the source and eventually to the ground, achieving a full closed circuit. Although this procedure is similar to the relay schematic and the MOSFET here is serving the same function as the relay. However, it can switch much faster than a relay. Besides, because there are no mechanical parts, it will reliably function for more switching operations than a relay. We decided to use a N-channel MOSFET in the Peltier control circuit, precisely the IRFZ44N [5] 5.4a is extremely low on-resistance per silicon area. This benefit is combined with the fast switching speed makes it suitable for our goal. Nevertheless, the MOSFET has more advantages for digital switching. The oxide layer between the gate and the channel prevents DC current from flowing through the gate, further reducing power consumption and giving a substantial input impedance. "The insulating oxide between the gate and channel effectively isolates a MOSFET in one logic stage from earlier and later stages, which allows a single MOSFET output to drive a considerable number of MOSFET inputs" [13]. This isolation also makes it easier for the designers to ignore to some extent loading effects between logic stages independently. That extent is defined by the operating frequency, as frequencies increase, the input impedance of the MOSFETs decreases. This feature combined with PWM 5.2.1 signal triggering at high frequency can assist in utilising the circuit as a switching voltage regulator [6]. Thus, the main purpose of the thermal control circuit in our implementation is to control the number of voltage passes to the Peltier element and eventually to control the heat's value. By looking back at figure 5.2, a 1KΩ resistor is serially connected in the path of the current going from a pin towards the MOSFET's gate. This resistor is necessary because of the maximum current specifications of the microcontroller. The manufacturer's absolute maximum ratings of the ATmega2560 specifies that the maximum current output per I/O pin is 40 mA, eventually 200 mA from all pins at the same moment.

[5] `http://bit.ly/MOSFET_IRFZ44N`

[6] `http://bit.ly/switching_voltage_regulator`



(a) IRFZ44N



(b) N-Channel MOSFET

Figure 5.4: IRFZ44N N-Channel MOSFET © International Rectifier

Also, it cannot supply more than 200mA from Vcc and GND pins. Moreover, the nominal voltage for the Arduino system is 5 V, and the Arduino does not limit the current to the 40mA max. Therefore, the pins will be damaged if the 40mA is exceeded and it is also very likely that the ATmega chip gets damaged as well. Nevertheless, the specification of the IRFZ44N status that the $R_{DS(On)}$ (Static Drain-to-Source On-Resistance) is 17.5mΩ. Using Ohm's Law as statued in Equation 5.1 to find the current of the entire circuit, which is the same amount anywhere within the series circuit.

$$V_{\text{Volts}} = I_{\text{Ampers}}/R_{\text{Ohms}} \tag{5.1}$$

Calculating the current flows in the circuit without adding any extra resistive element results in the following:

$$\begin{gathered} V = 5V \text{ and } R = 17.5m\Omega \\ I = 5V/17.5 \times 10^{-3} \approx 285A \end{gathered} \tag{5.2}$$

Therefore, a current limiting resistor in series with the MOSFET is needed so that the amount of current flowing through the circuit is controlled. A simple calculation based on the maximum current that the Arduino can supply results int he flowing:

$$\begin{gathered} V = 5V \text{ and } I = 40mA \\ R_{\text{Total}} = 5/40 \times 10^{-3} = 125\Omega \end{gathered} \tag{5.3}$$

Since the $R_{DS(On)}$ is very small, it is ineffective in relation to the total resistance of the circuit, as the total resistance in a series circuit is equal to the sum of the resistance in that circuit. Thus, any resistance from 125Ω and above can serve the purpose of current limiting to protect the microcontroller. However, as mentioned earlier, the maximum current that can be provided by the microcontroller across all the pins is 200mA. Hence, running 16 Peltier's control circuit still will damage the system. That is because the

Arduino will supply the whole system of 16 Peltier in parallel. In parallel circuits, the flow of electricity will be divided among all the paths according to the resistance along each of them. Thus, the voltage is the same across all paths, and the total current of the circuit is equal to the sum of the currents through each path as depicted in equation 5.4.

$$\begin{gathered} I_{\text{Total}} = \sum_{i=1}^{N} I_{\text{i}} \\ \text{Where N = Total number of paths} \end{gathered} \tag{5.4}$$

Respecting that each path will take around 40mA in case of a 125Ω resistor, the total current flow in the circuit will be:

$$\begin{aligned} I_{\text{Total}} &= \sum_{i=1}^{16} 40mA \\ I_{\text{Total}} &= 640mA \end{aligned} \tag{5.5}$$

Therefore, a higher resistance value is needed to stay within the 200mA limit of current supply. Using the same equation inversely to calculate the amount of current per path considering the total current is 200mA and equivalent resistance per path results in:

$$\begin{gathered} I_{\text{per path}} = I_{\text{Total}}/N \\ \text{Where N = Total number of paths} \\ I_{\text{Total}} = 200mA \text{ and } N = 16 \\ I_{\text{Path}} = 200/16 = 12,5mA \end{gathered} \tag{5.6}$$

From the current resulted in Equation 5.6 one can recalculate the minimum current limiting resistance needed. Using Ohm's Law, we get:

$$\begin{aligned} V &= 5v \text{ and } I = 12.5mA \\ R_{\text{Path}} &= 5/12.5 \times 10^{-3} = 400\Omega \end{aligned} \tag{5.7}$$

Although, the minimum suitable resistance to protect the circuit is 400Ω. However, all resistors are subject to changes in temperature. The effect of temperature variations depends on the type of construction and is known as the *temperature coefficient* [7], which indicates the relation between the resistor's value and temperature changes which may be either positive or negative. In general, composition resistors have a negative temperature coefficient and

[7] http://bit.ly/Temperature_coefficient

this means that composition resistors will decrease in resistance with an increase in temperature. However, the variation in a resistor per degree of temperature is linearly correlated to the temperature coefficient of that resistor. A high temperature coefficient indicates that the change in resistance is high. Furthermore, running the microcontroller at the absolute maximum current all the time will affect the microcontroller's lifetime and eventually affect the system's durability. Thereby, a higher value for the resistance is considered to operate the system at a reasonable rate. A 1kΩ happens to be a reasonable value for the circuit to control the current which will not damage the microcontroller and still can trigger the MOSFET's gate. With 1KΩ, the current will be 5mA and 80mA per path and in total respectively as illustrated in Equation 5.8. Thus, one can assure that the microcontroller will operate comfortably.

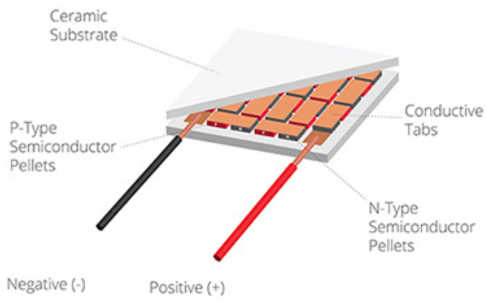
$$
\begin{aligned}
&V = 5V \text{ and } R = 1k\Omega \\
&I_{\text{Path}} = 5/1 \times 10^3 = 5mA \\
&I_{\text{Total}} = \sum_{i=1}^{16} 5mA = 80mA
\end{aligned}
\tag{5.8}
$$

Peltier

The Peltier effect is the presence of heating or cooling at an electrified junction of two different conductors [31]. More precisely, the effect creates a temperature difference by transferring heat between two electrical junctions. The Peltier effect named after its founder the French physicist Jean Peltier [8]. In his experiment, Jean Peltier observed that the heating or cooling of the junctions in a composite circuit of metals is directly proportional to the direction of the electric current passing around the circuit. Thus, the electric current created by applying a voltage across joined conductors flows through the junctions of the two conductors, cooling one junction and heating the other. This happens by removing heat at the junction in the direction of the current, and cooling occurs, while heat is deposited at the other junction [109]. Furthermore, thermoelectric modules 5.5b form an array of alternating n- and p-type semiconductors. These semiconductors which are of different types have complementary Peltier coefficients. The array of elements is soldered between two ceramic plates. In order to make a typical heat pump, multiple junctions are created between two plates. The multiple junctions between the two plates are soldered electrically in series in which current is driven and thermally in parallel in which heat is directed.

[8] `http://bit.ly/Jean_Peltier`

(a) TEC SP1848-27145



(b) Schematic view

Figure 5.5: Peltier modules © TEC

The Peltier effect generated at the junction per unit time is illustrated in Equation 5.9.

$$\begin{gathered}\dot{Q} = (\Pi_{\mathrm{A}} - \Pi_{\mathrm{B}})I \\ \mathit{Where}\ (\Pi_{\mathrm{A}}, \Pi_{\mathrm{B}}) \text{ are the Peltier coefficients of A and B} \\ \text{and } I \text{ is the electric current from A to B}\end{gathered} \tag{5.9}$$

One of the most popular applications of the Peltier effect are thermoelectric coolers. Although the main application of the Peltier effect is cooling, the Peltier effect can also be used for heating or control of temperature. In these cases, a DC voltage is required. However, the Peltier modules can also be used to generate DC voltage power by reversing the Peltier operation. When heat is directed towards one side and a cold material towards the other, the temperature difference between the two side will generate an electrified junction. However, several modules are needed to generate a slight amount of voltage, which is also applicable when using the modules for cooling. The cooling effect of any unit using thermoelectric coolers is proportional to the number of coolers used [109]. Typically multiple thermoelectric coolers are connected side by side and then placed between two metal plates. For the thermotactile interface, sixteen thermoelectric Peltier modules are used. Those are specifically of type TEC SP1848-27145) [9]. The main feature of this version that it is designed for heat pumping applications. As the semiconductors are refined in way to make less heat loss due the Peltier effect. Moreover, it is relatively lightweight and of a small size compared to other modules. Such that it is formed of 4x4x3.4mm as illustrated in Figure 5.5a. The surface is

[9] `http://bit.ly/TEC_Peltier_Module`

Temperature (degrees°C)	20	40	60	80	100	150
Open circuit voltage (V)	0.97	1.8	2.4	3.6	4.8	6.4
Current (mA)	225	368	469	558	669	895

Table 5.1: TEC SP1848 specifications

ceramic made of *bismuth telluride* [10] which is considered as the most suitable material for thermoelectric modules [109]. The module can operate between −40 °C and 150 °C, consuming different voltages as illustrated in Table 5.1. Nevertheless, it is obvious from the table that the relationship between the temperature and the voltage is non-linear, it is almost of exponential nature. For example, at 2.4V the temperature is around 60 °C degrees. However, in practice, it appeared that 2V is sufficient to reach 60 °C. Therefore, a careful control is needed to keep the heat value within an acceptable, unharmful range to the user's hand. Also, it is observed that the current consumption is almost the double of what is stated in the specifications. As the power supply used to feed the circuit is 12V/100A, the current consumption of the full sixteen Peltier modules is still within an acceptable value. More importantly, is the voltage control to keep the heat value around maximum 60 °C. Therefore, the thermal control circuit 5.2 is crucial to this implementation. As mentioned earlier, the circuit behaves as a voltage regulator to regulate the voltage from 12V down to 2V maximum. The fast switching mechanism of the MOSFET is recruited for this matter in combination with a PWM 5.2.1 signal. As the switch turns on/off at a very high frequency, the on-time rate determines the amount of the output voltage based on the 12V input. Finally, Figure 5.6 depicts one row of the final thermal circuit control. The same schematic applies to sixteen elements in four rows. However, only four elements are shown in the figure for the clarity of the scheme.

5.1.2 Haptic Interface

The haptic part is made out of sixteen haptic motors which are linear resonant actuators. Each motor is joined with a haptic driver 5.1.2 that functions as a sub-microcontroller to the LRA motor 5.1.2. However, the driver is not just for control, but, it also works as a local memory to store the waveform to generate the haptic feedback. Thereby, each data value will be mapped to distinct haptic feedback. The electronic circuit to control a single Haptic motor is illustrated in 5.2 and consists of an Arduino Mega as a microcontroller, a haptic driver and an LRA motor. However, as presented later, the

[10] `http://bit.ly/Bismuth_telluride`

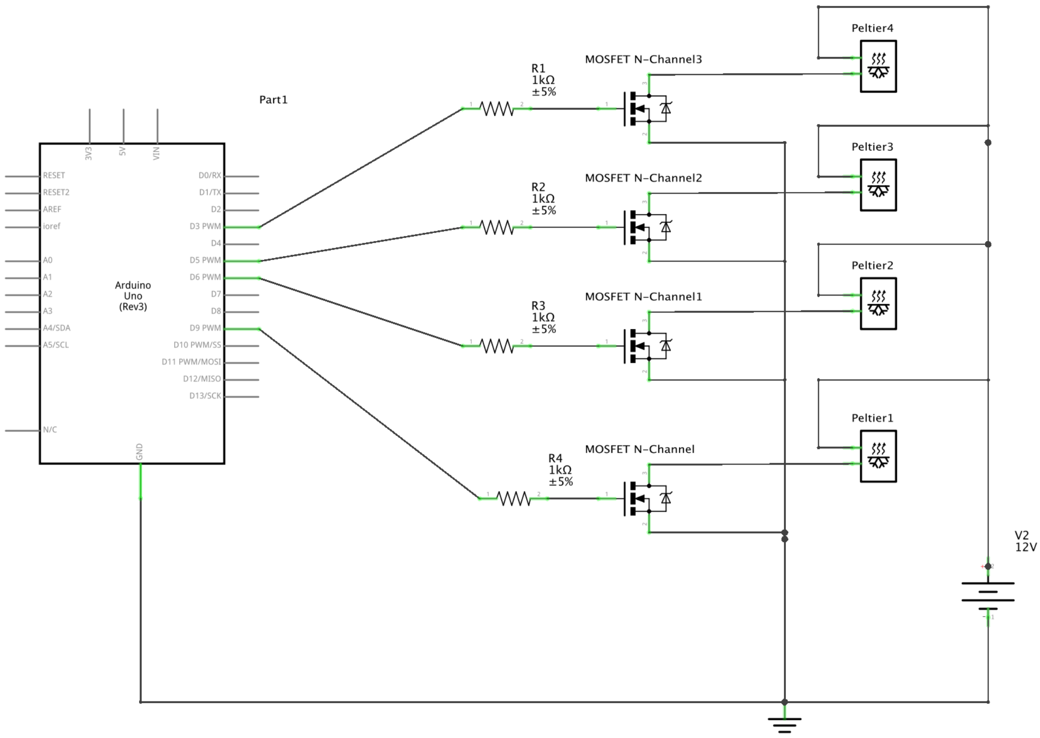


Figure 5.6: One thermal row control circuit

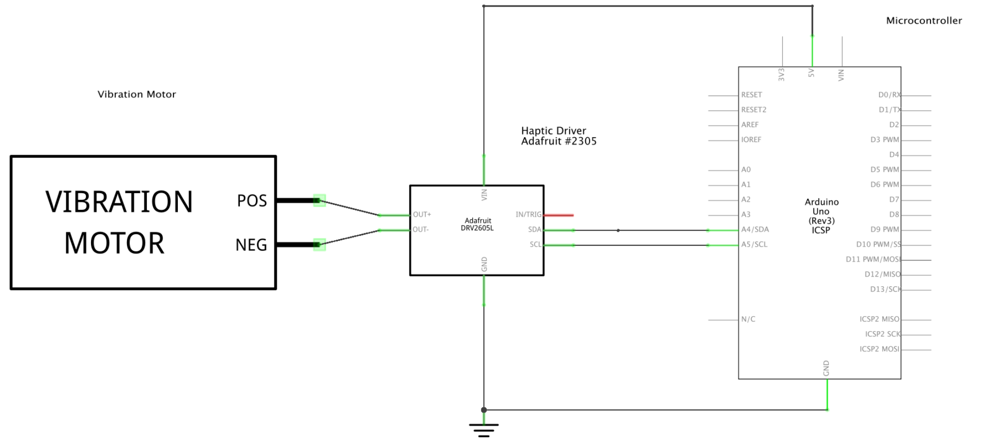


Figure 5.7: Single haptic control circuit

circuit will have a slight change when scaling up to sixteen motors. The reason behind is that the haptic drivers run over an I^2C bus 5.2.2, which requires a unique address to establish the channel. Unfortunately, due to the fast manufacturing demand, the manufacturer produced all the haptic drivers with the same address. Therefore an additional multiplexer 5.1.2 is needed to control the channel. For the rest of the circuit, the logic is the same for all the haptic motors. In the following subsections we detail each element of this electronic circuit.

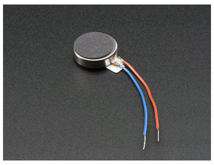
LRA Motors

LRA 5.8a stands for a linear resonant actuator, which describes the motion of an internal magnetic mass attached to a spring. LRAs are an alternative to eccentric rotating mass vibration motors (ERM). They work on a similar principle, where a moving mass creates an unbalanced force. When attached to an object, the force causes it to displace, which is what is felt as vibrations. The key difference between the devices is how they generate the force. While ERMs rotate an unbalanced mass to create a centripetal force, LRAs have no external moving parts. An electrical signal through the LRAs coils 5.8b will force the mass up and down, resulting in a force that causes displacement. The mass and spring system is a basic example of a simple *harmonic oscillator* [11], which demonstrates resonance. This means that the device oscillates with greater amplitude at particular frequencies. The mass is restricted to move in only one axis, unlike the ERM. This linear resonant behaviour is

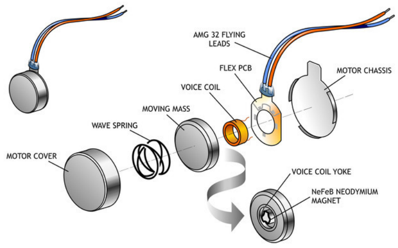
[11] `http://bit.ly/Harmonic_oscillator`

where the device gets its name. As the magnetic mass is driven in each direction, the LRA requires an AC signal, unlike ERMs which are based on DC motors. One may recognise LRAs are a similar concept to audio speakers, which are designed to have a balanced response across a range of frequencies. Otherwise, a particular tone would stand out in the music. However, LRAs are designed to be as efficient as possible. Both the bandwidth and the energy loss are described by the quality factor [12] (Q). A low Q factor indicates a wide bandwidth with a higher rate of energy loss. Therefore, LRAs have high Q factors, indicating a low rate of energy loss but a very narrow bandwidth. As a result, LRAs will only vibrate when driven within a few hertz of their quoted resonant frequency. Therefore, LRAs must be driven at their resonant frequency to generate a significant amount of force for a large current. Thereby, LRAs are suitable for haptic applications within a specific frequency range. Furthermore, specialised LRA driver chips are available such as the DRV2605L 5.1.2, which makes integrating LRAs much simpler, and offers a range of dedicated features. For example, the DRV2605 from Texas Instruments can drive ERMs or LRAs and has an automatic resonant frequency detection algorithm. Although they require a specialised driver, LRAs have a couple of key benefits over their ERM counterparts especially when it comes to implementing haptic feedback. A dedicated driver is often used in haptic feedback applications even when using an ERM, so there is no additional component cost. LRAs offer an excellent haptic response, meaning they start to vibrate very quickly and can be stopped quickly too. In fact, the normal stop time of an LRA can be significantly longer than an ERM motor, since an LRA can take up to 300ms to stop vibrating due to the continued storage of kinetic energy in the internal spring during operation. However, an active braking mechanism is usually used for an LRA. By performing a 180-degree phase shift of the AC signal provided to the actuator, the vibration can be stopped very quickly within approximately 10ms, which happens by producing a force opposite to the oscillation of the spring. The braking mechanism is embedded in the haptic drivers. This fast triggering feature makes an effect like a click to simulate a button press feel realistic, or even triggering a wave sequence with different stops can provide a wide range of distinct haptic forms. This is unique to LRAs as they can reproduce a precise waveform of varying intensity over time with a fixed frequency, whereas a waveform of varying intensity in an ERM motor will also produce a different frequency of vibration. Nevertheless, with no external moving parts, they are easily mounted in small, space-restricted enclosures like in smartwatches and other wearables. The main source of failure in DC motors is the precious

[12] `http://bit.ly/Q_factor`



(a) LRA Motor © Adafruit



(b) Schematic Overview © Precision Microdrives

Figure 5.8: Linear resonant actuator

metal brushes that wear out as the motor rotates. As LRAs do not suffer from this mechanical wear, they have much longer lifespans. However, due to their design, LRAs are limited in vibration amplitude. Based on all the enhanced features of LRAs and their significance advantages comparing to ERMs, it was evident to choose LRAs as the vibration motor in the implementation of the thermo-tactile interface. Depending on the fast triggering mechanism assisted by the DRV2605 driver, it was possible to generate distinct haptic feedbacks with a fixed frequency. This can eventually provide a convenient stimuli to the user while exploring a specific data point.

Haptic Driver DRV2605L

The DRV2605 5.10b from Texas Instruments [13] is a little motor driver. However, rather than controlling a stepper motor or DC motor, its designed specifically for controlling haptic motors, buzzers and vibration motors. Normally one would turn those kinds of motors on and off, but this driver can have various effects when driving a vibe motor. The driver has an extensive integrated library that has more than 100 effects. This library facilitates the process of generating waveforms, as one can rely on the existed effects to generate a combined sequence. Moreover, it has a licenced library from Immersion [14], specifically the TouchSense2200 software, which includes 2200 effects library and audio-to-vibe features. So that, with such a driver ramping the vibration level up and down, clicking effects, different buzzer levels, wave-sequence vibration or even having the vibration follow a musical, audio

[13] `http://www.ti.com/lit/ds/symlink/drv2605l.pdf`
[14] `https://www.immersion.com`

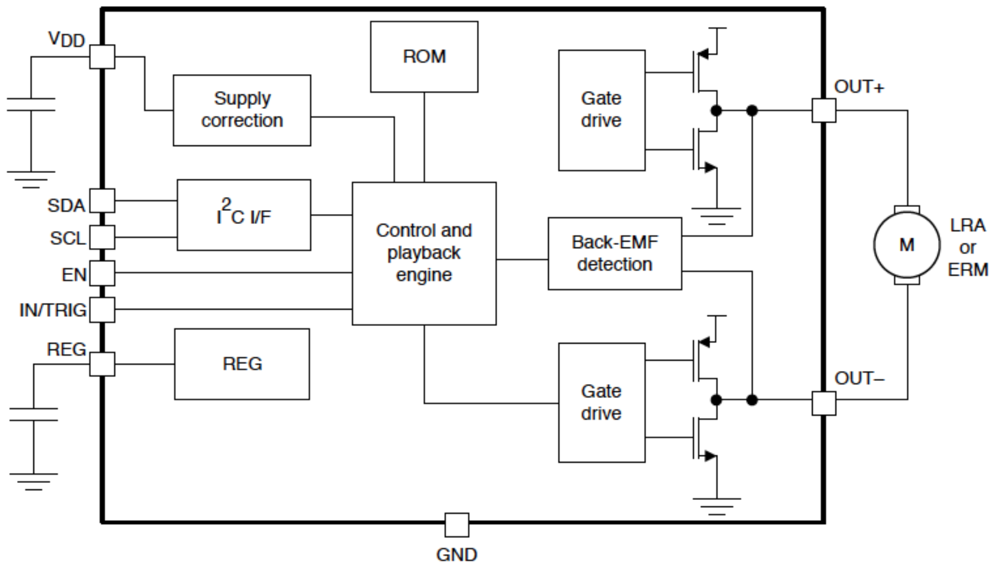
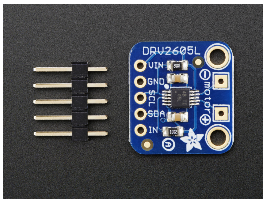


Figure 5.9: DRV2605 Functional Diagram © Texas Instruments

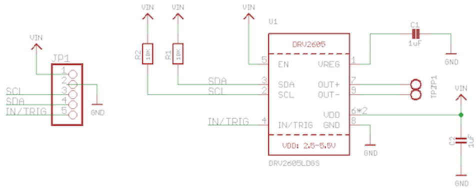
input or a PWM signal, are all possible. Furthermore, the driver is flexible to control both types of vibration motors (LRAs, ERMs). The smart-loop architecture inside the DRV2605 device provides an easy auto-resonant drive for LRA, as well as feedback-optimised ERM drive allowing for automatic overdrive and braking. These features create a simplified input waveform paradigm as well as reliable motor control and consistent motor performance. The chip is controlled over an I^{2}C bus, which in turn relieves the host processor from generating the signal repeatedly. Eventually, it will save the timer interrupts and hardware pins. After initialisation, a string of multiple effects can be strung together in the chip's memory and then triggered to actuate in a row. The chip has eight internal registers acts as waveform sequencer queue. These eight sequence registers can store up to eight waveforms for sequential playback. Figure 5.9 illustrates the general functionality of the driver.

For the simplicity in the implementation of the thermo-tactile interface, a chip configured onto a breakout board is used 5.10a. Adafruit [15] makes this breakout board. It works with both 3V and 5V power logic, and they also provide some code specifically for Arduino. However, their library was limited for starter projects. Therefore, it was necessary to reimplement the library from scratch, following the specifications from the datasheet. As

[15]`https://www.adafruit.com/product/2305`



(a) DRV2605 Breakout © Adafruit



(b) DRV2605 Schematic overview © Texas Instruments

Figure 5.10: Haptic Driver

illustrated in Figure 5.10a, the DRV2605 has the following 5 pins:

- Vin: To power the board. Recommended to give it the same power as the logic level of the microcontroller (e.g. 5V for a 5V microcontroller like Arduino).
- GND: A common ground for power and logic
- I^{2}C pins:
  - SCL: I^{2}C clock pin, the pin can be used with 3V or 5V logic, and there is a 10K pull-up resistor on this pin.
  - SDA: I^{2}C data pin, also can be used with 3V or 5V logic, and there is a 10K pull-up resistor on this pin.
- IN/TRIG: This is a general purpose pin that can be used for a couple of different uses. Such as reading an analog audio signal to control the audio-to-haptic code, or to trigger the effects to go rather than sending an I^{2}C command. Finally, it can be used for real-time signals to haptic or to trigger the motor based on a PWM signal.

The thermo-tactile interface uses the DRV2605 as a haptic control driver as illustrated in Figure 5.7. The haptic driver in this logic is connected to the main microcontroller over an I^{2}C bus. It behaves as a sub-microcontroller to trigger the vibrator. Also, the internal waveform sequencer queues are used to store the desired sequence for playback, based on the value of the data retrieved from the dataset. Finally, the GO register is used continuously throughout the operating period of the system. This register is linked with the data coming from the hand tracking camera, which runs continuously to track users' hand and the intended position of exploration. As the user's hand hovers over a haptic position, that motor will be activated. Otherwise, it is idle. Thus, the haptic driver is used to provide a switching mechanism as well by altering the triggering bit inside the GO register, Which in turn provides a kind of interactivity to the system. Finally, the only limitation in the DRV2605 is the fixed I^2C address of the device which is 0x5A. This inconvenient limitation is due to the speed of manufacturing and to keep the device at an affordable price. However, for scaling the thermo-tactile interface up to sixteen functional motors, an additional multiplexer is assisted in controlling the communication channel between the main microcontroller and the identical haptic driver ends.

TCA9548A Multiplexer

The TCA9548A 5.12 from Texas Instruments [16] is an eight channel multiplexer. This multiplexer has eight bidirectional gates that can be controlled over an I^2C bus. Using the TCA multiplexer is fairly straightforward. The multiplexer itself is on I2C address 0x70. However, its address is also adjustable from 0x70 to 0x77, which means that the TCA its self is scalable. Since one could have 8 of these multiplexers on each of the 0x70 to 0x77 addresses to control 64 of the same I^2C addressed terminal devices. Moreover, it behaves as a gatekeeper such that switching between the eight downstream pairs through single SCL/SDA pins from the main microcontroller, any individual SCn/SDn channel or even combination of channels can be selected. The programmer has to send a single byte with the desired multiplexed output number and the gate will be initialised. A single programmable control register controls this mechanism. Thereby, the TCA9548A can also be used for broadcasting. However, the main application for this type of multiplexers is to control the channel between the main microcontroller and a terminal sensor or actuator. Usually, these sensors or actuators are manufactured with a fixed I^2C. Thereby, it is not possible to control them independently in case of several identical sensors or actuators are attached to the same microcontroller. In our application the multiplexer is employed exactly for this purpose, to control multiple haptic drivers of the same address. Again for the simplicity in the implementation of the thermo-tactile interface, a chip configured onto a breakout board is used 5.11. Adafruit [17] makes this breakout board. The chip itself is 3V and 5V compliant so that it can be used with any logic level. Another nice feature in the TCA9548A is the ability to limit the maximum high voltage, which allows the use of different bus voltages on

[16] http://www.ti.com/lit/ds/symlink/tca9548a.pdf
[17] https://www.adafruit.com/product/2717

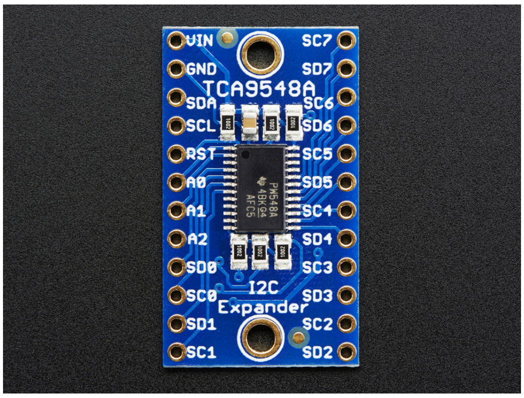


Figure 5.11: TCA9548A breakout © Adafruit

each channel, so that 1.8V, 2.5V or 3.3V terminal devices can communicate with a 5V device without any additional protection.

- Vin: Voltage Input pin to power the board, accepts 3 and 5V. Recommended to give it the same power as the logic level of the microcontroller (e.g. 5V for a 5V microcontroller like Arduino).
- GND: A common ground for power and logic.
- I^2C pins:
  - SCL: I^2C clock pin, the pin can be used with 3V or 5V logic and there is a 10K pull-up resistor on this pin.
  - SDA: I^2C data pin, can also be used with 3V or 5V logic and there is a 10K pull-up resistor on this pin.
  - RST: The reset pin, for resetting the multiplexer chip. This pin is pulled high by default, so it should be connected to the ground to reset.
- A0, A1, A2: These are the address selection pins for the multiplexer. By default, the multiplexer is at address 0x70 and these three pins are pulled low in the Adafruit breakout version. Therefore, they must be connected to Vin, to set the address to 0x71 up to 0x77. For this to be achieved, external soldering on the breakout board should be placed.
- I2C Multiplexed-Side pins:
  - SDx and SCx: SDx for I^2C data and SCx for I^2C clock. There are eight sets of SDx and SCx pins, from SD0/SC0 to SD7/SC7.

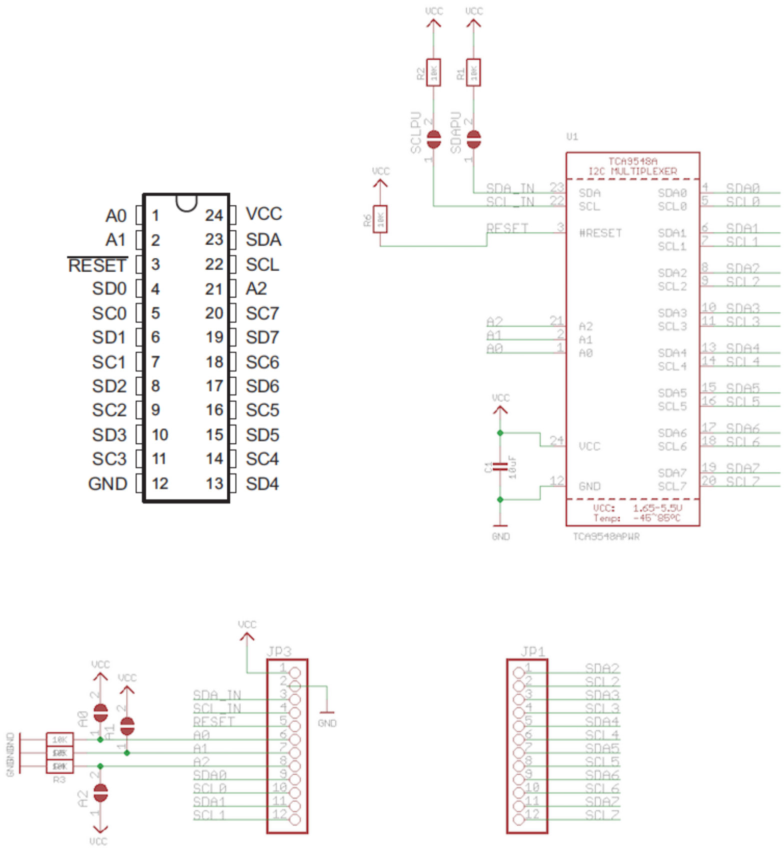


Figure 5.12: TCA9548A schematic overview © Texas Instruments

These are the multiplexed pins. Each one is a completely separate I^2C bus set.

Moreover, Figure 5.13 depicts the list of the possible address in the TCA9548A chip based on altering A0, A1 and A2 pins. In case of our implementation, extra solder is added to the A0 pin to get the address 0x71. Since two multiplexers are used to control sixteen haptic motors, each connected to eight. Thereby, one microcontroller is enough to control the haptic part. Finally,

INPUTS			I^2C BUS SLAVE ADDRESS
A2	A1	A0	
L	L	L	112 (decimal), 70 (hexadecimal)
L	L	H	113 (decimal), 71 (hexadecimal)
L	H	L	114 (decimal), 72 (hexadecimal)
L	H	H	115 (decimal), 73 (hexadecimal)
H	L	L	116 (decimal), 74 (hexadecimal)
H	L	H	117 (decimal), 75 (hexadecimal)
H	H	L	118 (decimal), 76 (hexadecimal)
H	H	H	119 (decimal), 77 (hexadecimal)

Figure 5.13: TCA9548A address reference © Texas Instruments

Figure 5.14 shows two rows of the final haptic circuit control with one multiplexer. The same schematic is applicable to the sixteen elements in four rows controlled by two multiplexers. However, only the output of one multiplexer is shown in the figure for the clarity of the scheme.

5.2 Software

As discussed in the previous section, the thermo-tactile system consists of two parts, each of them has its own independent electronic circuit and logic and an individual microcontroller is mapped to each part. The microcontroller is responsible for processing different types of data and providing a specific controlling signal to the circuit attached to it. The thermal part is controlled by a PWM signal 5.2.1 to tune the amount of voltage passing to the Peltier element. On the other hand, the haptic part is controlled through an I^2C bus 5.2.2, using byte messages to update, drive and trigger the motor. Both microcontrollers are Arduino Mega2560 and they are by default C++ compatible. Although, it was possible to combine all the logic in one microcontroller, each part is being controlled on a different communication medium. However, the separation of concerns is necessary to maintain the scalability of the whole system. Also, as mentioned earlier, this separation gives the flexibility to reform, alter and reconfigure the system in the future.

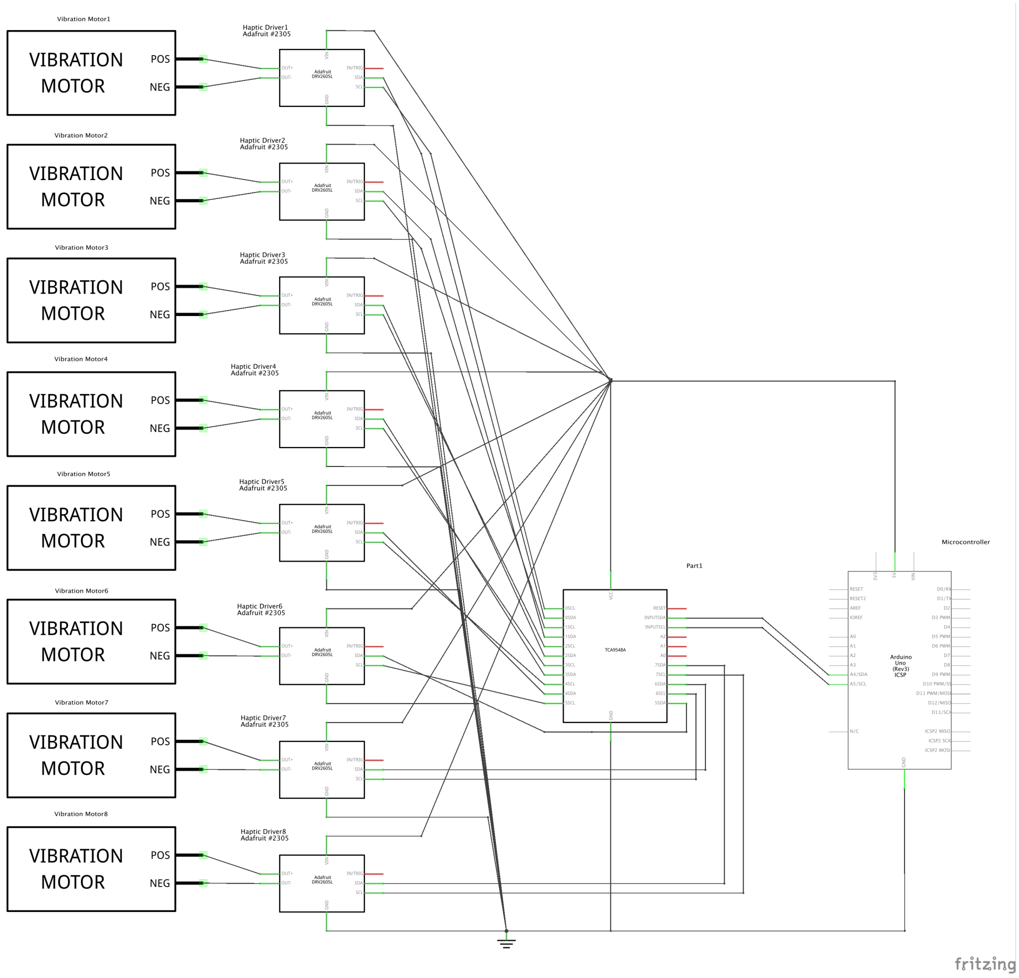


Figure 5.14: Two haptic rows controlled by a single multiplexer

Moreover, the Arduino microcontrollers do not support multithreading or multitasking architecture.The only way to achieve such an architecture is by manually implementing a scheduler based on clock timing and delays. Such an implementation adds extra complexity and may effect the signals on the I^2C bus in term of delivery and latency, if for example a signal is sent from a haptic slave to the master while the master is on delay for a PWM control signal. Thus, running all the logic on a single microcontroller will adversely affect the total performance. Instead, the interfacing between the two parts of the system together with the data to be represented is happening on a traditional computer acting as a brain. The system is therefore similar to a *Multilayer Architecture* [18]. Nevertheless, deploying the system in such a way, enhances the simplicity, scalability and the reusability goals of the system. Thereby, adding more parts or even more logic in form of layers can happen smoothly, this simplicity has been materialised by adding a hand tracking feature to the system. The hand tracking system consist of a traditional webcam for tracking and a detection logic implemented, based on the computer vision library (OpenCV) [19] 5.2.3. Moreover, the hand tracking system runs independently from the thermo-tactile system. However, they are both running on the same brain computer. Therefore, only a simple python script is written to interface the communication between the hand tracking and the themo-tactile systems.

5.2.1 Thermal Interface

The general overview of the logic implemented for the thermal part of the system is simple as illustrated in the class diagram 5.15. The code A.1 consist of three arrays, heat values setup function and PWM setup and control functions. The first array is to store the number of a pin to which a thermoelectric control circuit is attached, the second to store the dataset values and the third to store the heat values to be rendered later by the Peltier elements. The data values can be hard-coded inside the program or passed through the serial port at the beginning of the program. The heat values setup function is implementing a simple linear vector scaling [20] method, to map the values from the range of the dataset to the range of the PWM. Equation 5.10 illustrates the mathematical method behind the mapping function

[18] `http://bit.ly/Multitier_architecture`
[19] `https://opencv.org`
[20] `http://bit.ly/Linear_map`

and Figure 5.16 shows the process.

$$\begin{aligned} \vec{V}_1 &= (v_1 - R_{1\mathrm{L}}) \\ \vec{V}_2 &= X \times \vec{V}_1 \\ X &= (R_{2\mathrm{U}} - R_{2\mathrm{L}})/(R_{1\mathrm{U}} - R_{1\mathrm{L}}) \\ v_2 &= \vec{V}_2 + R2L \\ \text{Therefore}, v_2 &= (\frac{(v_1 - R_{1\mathrm{L}}) \times (R_{2\mathrm{U}} - R_{2\mathrm{L}})}{R_{1\mathrm{U}} - R_{\mathrm{L}}}) + R_{2\mathrm{L}} \end{aligned} \tag{5.10}$$

The vector $\vec{V}1$ is the vector between the lower bound of the data range and the data point $v1$. While the vector $\vec{V}2$ is the output vector resulted from multiplying the factor X by the vector $\vec{V}1$. The scaling factor X is the ratio between the two ranges. Thus, the mapped value $v2$ is the resulted value from the placement of the vector $\vec{V}2$ on the desired range (i.e. the PWM range in our implementation), shifted by the lower bound of that range. This shifting is necessary to generalise the formula to any desired range, since the output vector $\vec{V}2$ is starting from zero, therefore the placement has to be shifted to match the bounds of the desired range. As a result, the lower bound of $\vec{V}2$ will be equal to the lower bound of range 2 (the desired range) and the upper bound of $\vec{V}2$ is the mapped value. This mapping mechanism is necessary in our application, as the goal is to represent different types of data. These data can be of any range, while the range of the PWM control signal is between (0 – 255) by default.

PWM

Pulse-width modulation (PWM) is a modulation technique used to translate a message into a pulsing signal [13]. PWM is used in different applications, mainly in controlling the power of electronic devices and inertial loads like motors (i.e. servo motors) in varying intensities and speeds. However, it can also be used to encode transmission's information over a communications channel. The concept of PWM is that the voltage and current supplied to a load is controlled by an iterative switching (On/Off) between the load and the power at a very fast frequency rate. The output of this switching mechanism is the average of the supplied voltage, based on the duty cycle. The duty cycle 5.11 is the pulse width of the On-time per cycle from the output frequency and it is represented in percent (100%). Therefore, a higher duty cycle results in a higher total average power supplied to a device, which makes the PWM suitable for voltage regulation circuits, as varying the duty cycle results in different voltage output. Equation 5.11 illustrates the mathematical concept of the PWM, and figure 5.17 shows the PWM duty cycles.

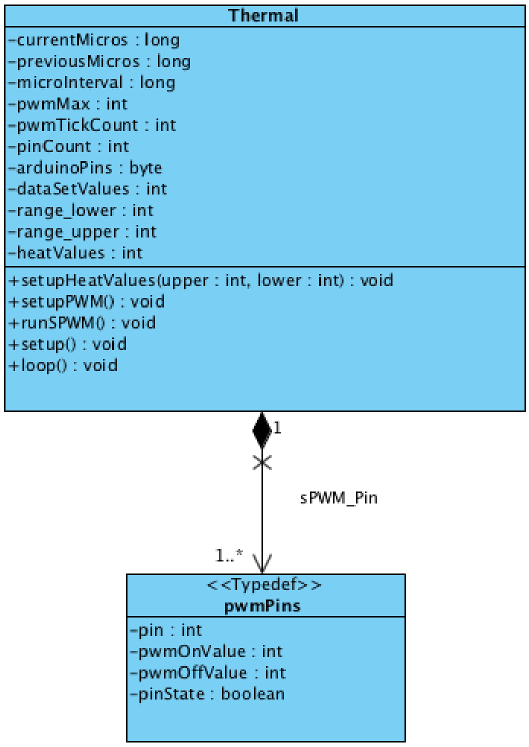


Figure 5.15: Thermal class diagram

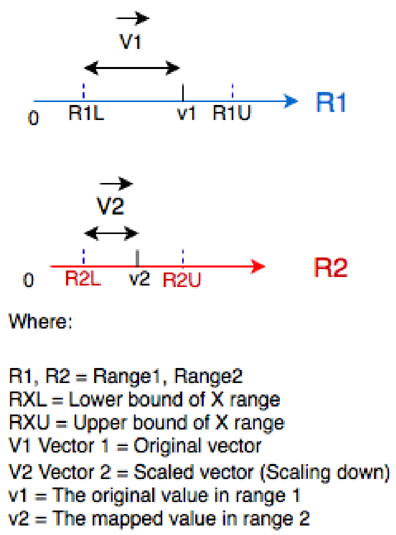


Figure 5.16: Linear vector mapping

$$\begin{aligned}\text{Frequency} &= \frac{\text{Cycles}}{\text{Second}}\\ \text{Duty Cycle} &= (\frac{Pulse\ Width_{(\text{On-time})}}{\text{Cycle}}) \times 100\\ V_{\text{OUT}} &= \text{Duty Cycle} \times V_{\text{IN}}\\ V_{\text{OUT}} &= (\frac{\text{On time}}{\text{Max time}}) \times V_{\text{IN}}\end{aligned} \tag{5.11}$$

Furthermore, usually a low pass filter is used to filter out the noise of the switching mechanism due to low PWM frequency rates. Thereby, the frequency of a PWM signal must be high enough to provide fast switching that is able to keep the load in a steady state in which the signal is perceived smoothly by the load. Otherwise, if the noise is so high, a fluctuation in the power will occur that might damage the device. Therefore, the frequency of a PWM signal is dependent on the load of the application. For example, a lamp requires at least 120Hz frequency to light dimly, while driving a motor requires between few kilohertz up to tens of kilohertz, whereas hundreds of kilohertz is required for audio amplifiers. Furthermore, a significant advantage when using PWM is that power loss is very low in the switching devices . That is because when the switch is Off, there is practically no current and when it is On the power will be transmitted to the load and there is almost no voltage drop across the switch. Thus, the power loss in both cases is almost zero, or very close to zero. The power loss defined by Robert Boylestad as "*If a current I flows through through a given element in a circuit, losing voltage V in the process, then the power dissipated by that circuit element is the product of that current and voltage:* $P = I \times V$".

The main purpouse for the PWM in our application is to control the MOSFET's gate as to provide switching and voltage regulation mechanism. The Arduino Mega contains fifteen pure PWM pins by default, and the PWM resolution is one byte resulting in (0 – 255) values range. However, in our application, sixteen PWM signals are needed to control Peltier elements. Also, the Arduino executes the PWM signals sequentially in case several subsequent PWM signals exist in the program, and waits until a ready-status message is returned from the ATmega2560 microcontroller after each PWM instruction. Thus, iterating over each pin will result in a bad performance, as each Peltier has to stay Off for a period equivalent to fifteen times its received signal time. Eventually, the output voltage will not be stable as a result to the gap between the control and Off-time. From this point, a software made PWM (SPWM) is implemented instead. The SPWM is by all means identical to the pure PWM. The only difference is that SPWM

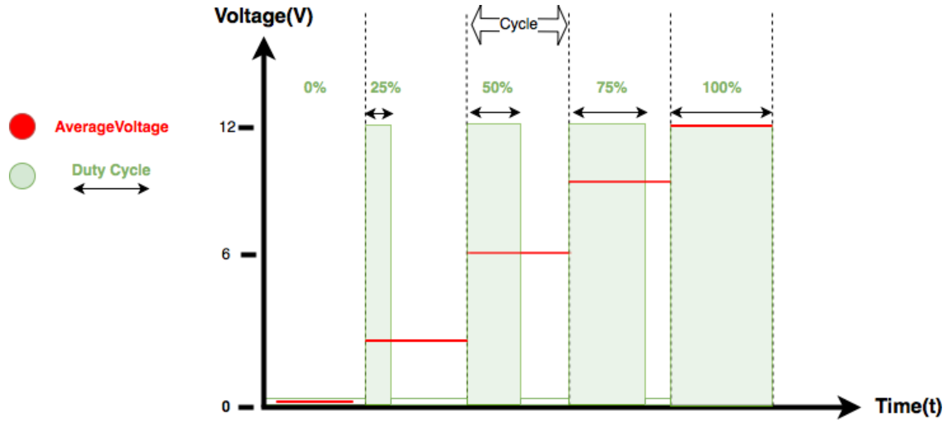


Figure 5.17: Pulse width modulation

signals are digital while the pure PWM is an analog modulated signal by nature. The SPWM produces digital HIGH and LOW (i.e. zero, one) pulses continuously in a varying successive time to simulate the duty cycling mechanism. However, there are three main advantages in using SPWM over PWM in our application. First, all the 54 GPIO pins can be used to generate a SPWM signal (i.e. all the digital pins), such that up to 54 Peltier elements can be controlled by a single Arduino Mega, which enforces our scalability goal. Second, the resolution and the frequency are flexible for modifications. Third, the way that we have implemented SPWM in which it makes iterative checks on the state of each pin and if pulse state should be updated. In this way, the SPWM signal will be updated instantly, if for example the duty cycling period is already passed. In other words, there will be almost zero delay in switching the state of the signal between HIGH and LOW. The result is a smooth and steady output signal. In our code, the void function *runSPWM()* 5.1 is responsible for checking and triggering the SPWM on each pin. Whereas, the other void function *setupSPWM()* is responsible for initialising the pins, their state, determining the On- and Off-time based on the mapped heat value. However, in this function an extra mapping is called to constrain the value of the SPWM to maximum 40, which is approximately 16% of the cycle time. The mapping is based on the same concept in Equation 5.10 and the value 40 is calculated in Equation 5.12. The reason behind this mapping is to limit the average output voltage passing to the Peltier element to less than 2 volts, which will limit the maximum rendered heat value to approximately 60 °C, which is somehow unharmful. Although 60 °C may also be a bit high to the user, for the forearm and fingertips it is considered very hot but not to a burning degree [56]. This evidence is based on Stevens and Choo's observations of the temperature sensitivity of the body. Stevens and Choo state that "*The ability to perceive changes in skin temperature depends on a number of variables including the location on the body stimulated, the amplitude and rate of temperature change, and the baseline temperature of the skin*" [56]. Moreover, Stevens and Choo state that the lips are the most sensitive area of the body, whereas the feet are the least sensitive. In the same context they described a difference for the hand in sensing cold and warm thresholds, as they are lower at the base of the thumb as compared to the forearm and fingertips. Nevertheless. the thermo-tactile surface is coated by a copper plate, this extra layer will cause a slight loss in the intensity of heat comparing to the Peltier's surface due to the diffusion of heat across the copper surface. Also, while testing the thermo-tactile interface, a 60 °C was presented and it felt natural hot but unharmful. Finally, the users will explore the haptic surface under a controlled setup, and precautionary measures will be taken to prevent any possibility of harm to the user.

```
void runSPWM()
{
  currentMicros = micros();
  //check if the cycle his finished (to increment PWM_ticks_counter)
  if (currentMicros - previousMicros >= microInterval)
  {
    //Set, the previous time to the current for the next cycle
    previousMicros = currentMicros;
    //loop over all the pins to check their cycle
    for (int i = 0; i < pinCount; ++i)
    {
      //set the state to ON as default
      softPWMpins[i].pinState = ON;
      //if the number of ticks reachs the needed pwm on
      //(evalute to true after several runs, when the specific pin has already made the amount of pwm_on
      // If true, then change the state of the pin to OFF
      if (pwmTickCount >= softPWMpins[i].pwmOnValue)
      {
        softPWMpins[i].pinState = OFF;
      }
      //Check if the tick counter completed already a full pwm cycle
      //If true, switch the state back to ON and reset the counter
      if (pwmTickCount >= pwmMax)
      {
        // softPWMpins[i].pinState = ON;
        pwmTickCount = 0;
      }
      // Write to the pin based on the state (On = High, Off =
```

```
    Low)
      digitalWrite(softPWMpins[i].pin, softPWMpins[i].pinState);
    }
    //increment the tick counter to start new tick.
    pwmTickCount++;
  }
}
```

Listing 5.1: Software PWM

$$\begin{aligned} On\ time &= (\frac{V_{\text{OUT}}}{V_{\text{IN}}}) \times Max\ time \\ On\ time &= (\frac{2}{12}) \times 255 = 42.5 \end{aligned} \tag{5.12}$$

5.2.2 Haptic Interface

The program's code of the tactile A.2.2 part of the system, is a mix of three logic levels. The first level concerns the logic of communications between the microcontrollers and the haptic drivers through multiplexers. The second logic level is responsible on the communication between the computer and the microcontroller. This level handles the data coming from the computer, such as the dataset values at the beginning of the run and the user's hand position throughout the running time. Even if the dataset values have been hard-coded in the script of the program, handling the position values will still function normally, as they are two independent functions. The third level is operational logic to bind the data values to the corresponding haptic driver and to control the triggering and braking of the LRA throughout the running time, based on the user's hand position. Figure 5.18 shows the general class diagram of the haptic implementation.

The communication between the computer and the microcontroller is happening over the serial port. With the help of a UART [21] chip embedded in the Arduino hardware, the microcontroller supports serial communication on pins 0 and 1, for electronic devices and also via a normal USB connection to the computer. The main advantage of a UART is that it allows the ATmega chip to receive serial communication even while working on other tasks, as long as a space in the 64 byte serial buffer exist. Moreover, the serial connection can be established on different rates, starting from 300bps up top 115 200bps. For the purpose of our implementation, the serial connection is established at 115 200bps rate for a maximum throughput. The built-in serial

[21] `http://bit.ly/UARTport`

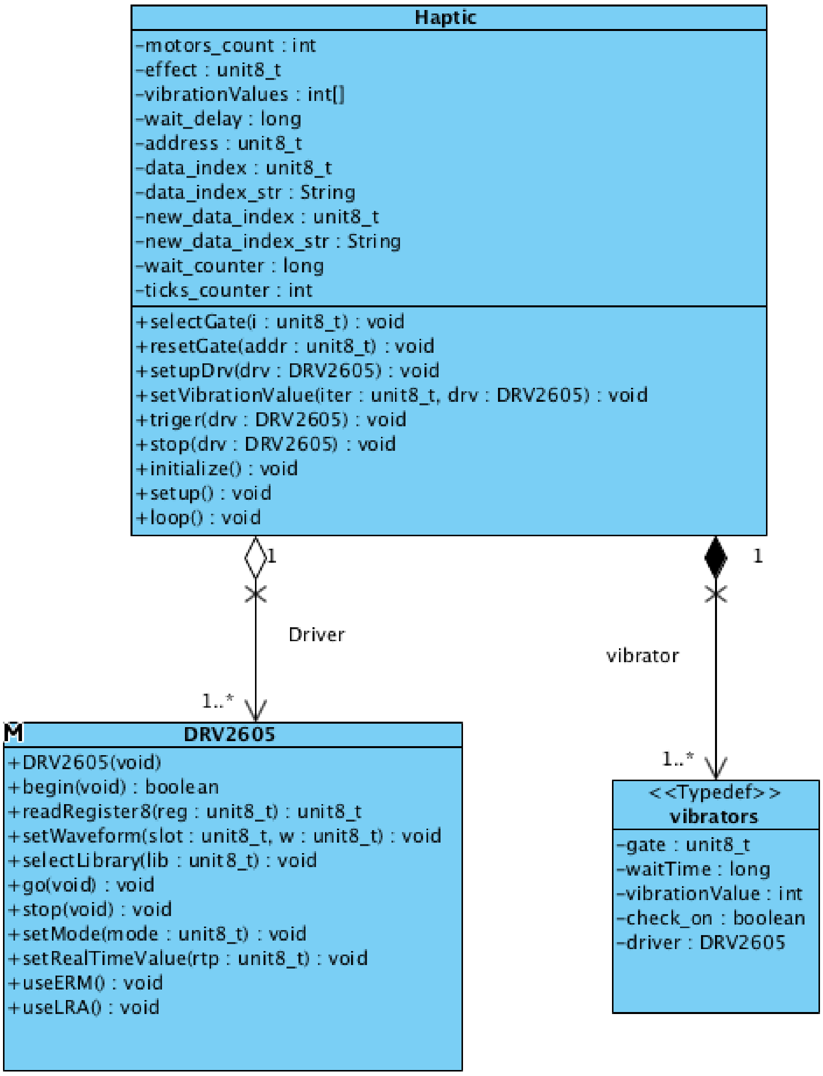


Figure 5.18: Haptic class diagram

library for Arduino is used to implement the code inside the microcontroller, which is actually a child class of the stream class. Whereas on the computer side, a python script A.3.2 is written using the python library PySerial [22] to handle the operations (e.g. send a data value, send a new position, or receive a print message). On the other hand, the communication between the microcontroller and the multiplexers, eventually to the haptic drivers, is operating over an I^2C bus 5.2.2. The microcontroller selects a gate from the multiplexer to pass data through to the driver.

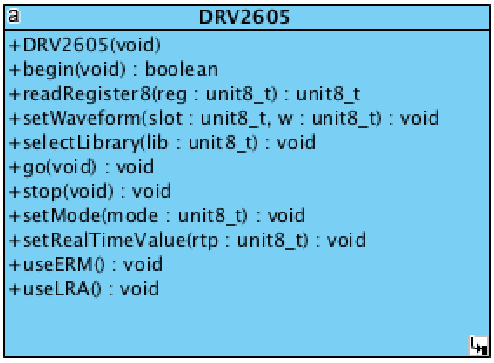


Figure 5.19: DRV2605 class diagram

The operational logic of the haptic drivers is implemented in two phases. The first phase is implementing a library A.2.1 that encapsulates the low level operations on the driver's domain. The library depicted in Figure 5.19 translates all the possible commands that the driver can accept for reusable functions, which in turn provides simplicity, reusability and separation of concerns in the code. Although, the basic functionality is provided by the manufacturer of the breakout board [15], a huge modification has to be done and a big part is added. The original library was limiting the use of the driver to ERM type motors by initialising the driver with fixed configuration. Also, not all the internal libraries of the driver were supported, neither all the modes, nor writing to the wave-sequencer. Adafruit's developers implemented that library as a starter point for hobbyists. Whereas in our implementation all the possible functionalities the driver chip can provide, is supported, but it might be more complex as one has to read the datasheet to understand different options provided. However, all the modifications and additions are based on the informations from the datasheet [13] of the chip provided by Texas Instruments. The header file A.2.1 defines all the internal registers, most of these were already defined by the original library, the

[22] `https://pyserial.readthedocs.io/en/latest/`

addition is the control registers and wave-sequencers. Regarding functions, reading and writing to any register is added so that a developer can manually modify any internal register configuration to the desired value. Moreover, setting a wave form is added to store a waveform in any of the 8 sequencer's registers. Furthermore, the set mode function is modified to accept any of the seven modes that the driver can operate and stop, triggering functionality is added. Finally, the DRV2605.cpp A.2.1 file implements all the defined functions in the header file using the defined variables. The second phase is inside the Arduino code, where a single instance of the haptic driver class is made. As all the haptic drivers are identical, one instance can control them all. Moreover, a functions is implemented to configure the driver at the beginning of the program (i.e. void setupDrv(DRV2605 drv)). After initialising the first connection to the driver, the 4th internal library is chosen as it has the strongest intensities of waveforms. The mode is set to the internal library mode, and 4x braking factor with a medium gain is set in the feedback register of the driver. Braking stabiliser and open loop waveform LRA auto resonant algorithm are set in the second and third control registers. Furthermore, another innitialise function is implemented to setup each motor based on the index in the grid of the 16 haptic motors void initialise. The setup is mainly mapping the vibration value to its motor, setting the waveform and number of ticks based on that value. It is important to mention that the haptic interface supports 5 distinct vibration feedback. The number is limited to 5 as after several trials, it conceived to us that a larger set of vibration may confuse the user and affects their ability to remember the differences between different vibration. Also, it will require explicit explanation about what each feedback means before the user is able to explore the data through. Therefore, five levels of vibration based on the same effect and vary in length are implemented. However, the number of times that a feedback is repeated is related to the level. In other words, a feedback of level one will be of short length and triggers once, whereas, a feedback of level five will be of 5 times the short length and triggers 5 times. Eventually, the developers cannot argue if that is good or bad before performing a user experiment, which can give some insight about how to improve the feedback form and set. Thereafter, when the setup of each motor is done, the program will be in a loop state, waiting for position updates to trigger the corresponding motor. Two independent functions are implemented for triggering and braking a motor.

I^{2}C

The Inter-Integrated Circuit (I^{2}C) is a synchronous single-ended [23] serial computer bus that supports packet switching and the multi-master, multi-slave archetype. The main use of this bus is to attach low-speed peripheral integrated circuits to microcontrollers in commonly short-distance, intra-board communication. An I^{2}C bus consist of two bidirectional lines. The serial data line (SDA), which is responsible for transferring data bits and the serial clock line (SCL) which is responsible on clock synchronisation between the connected chips and its signal is generated by the master device. Moreover, as the two lines are either open-collector or open-drain, usually they are pulled-up with resistors to prevent floating because the devices on the bus are active low by default (i.e. the pins are connected to the VCC through a resistor to prevent noisy readings by forcing the pin to be in the High-state) [83]. Moreover, the I^{2}C bus commonly operates over 5V or 3.3V as they are the most common voltage supplies for microcontrollers. Also, most of the I^{2}C versions are 7-bit addressing space allowing one bus to connect up to 128 uniquely addressed device. The bus operates on different speeds, based on the version and the mode (i.e. 10 kbit/s low-speed mode, 100 kbit/s standard mode, 400 kbit/s Fast mode, 1 Mbit/s Fast mode plus and 3.4 Mbit/s High Speed mode) [39]. The fast mode is the most common and it is the one that is used in our implementation. Furthermore, the I^{2}C protocol uses 8 bit-sequence signals to transfer data, respecting one-bit at a time. A typical I^{2}C message as illustrated in Figure 5.20 starts with a start condition, which occurs when a data line drops low while the clock line is still high. Thereafter, a 8-bits sequence is sent indicating the slave address with the most-significant bit indicating read or write message. Subsequently, another 8-bits sequence is sent indicating the sub-address of the internal register in the salve device. After that continuous 8-bits sequences of data are sent until a stop condition occurs. Nevertheless, each byte transferred on the bus is acknowledged by the receiving. Also, each transfer operation on the bus is initiated by the master with the start condition (i.e high-to-low transition on the SDA) and stopped by the master with the stop condition (i.e. low-to-high transition on SDA). Finally, the clock signal is high when a start or stop condition is being sent, whereas it is low for data-bit transmissions. In our application the Wire Library [24] is recruited to implement the communication logic over the I^{2}C bus. Although, the wire library has one limitation as it uses a 32 byte buffer for communication. However, the maximum number of bytes per transmission in our application is eight. Therefore, this barrier is of no ef-

[23] `http://bit.ly/Single-Ended-Signaling`

[24] `https://www.arduino.cc/en/Reference/Wire`

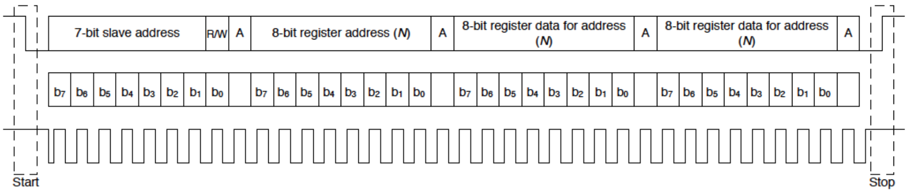


Figure 5.20: I²C Sequence © Texas Instruments

fect to our purpose. The tow multiplexers in our application are attached to the Arduino Mega controller on pins 20 (SDA) and 21 (SCL). Each one is then controlling the I²C gate of 8 haptic drivers. Therefore, two messages are sent each time the microcontroller has to communicate with a specific driver. The first message is sent to the multiplexer to select a gate specific gate. The multiplexer will initiate the gate and leave the bus back to the microcontroller. Thereafter, the second message is sent directly to the driver over the existing gate (i.e. setup, trigger or stop messages). Finally, a new I²C message will be sent to the multiplexer to reset the gate and return to standby mode. This resting is important in case that another driver has to listen and if the previously initiated gates are not reset, all the drivers on these gates will listen as in broadcasting modes. Eventually, the drivers will interfere each other resulting in wrong operations.

5.2.3 Hand Tracking (OpenCV)

The main purpose of using computer vision in our implementation is to provide a hand tracking functionality, which in turn brings some interactivity to the thermo-tactile interface. Also, without tracking the position of user's hand, one has to let all the motors triggering all the time, which will result in a bad exploration experience as the vibration feedback will interfere each other resulting in a confusing mix of vibrations. Nevertheless, as the motors have to trigger all the time, the durability of the device will be affected, since these vibration are made to operate occasionally and running them constantly for a long time will damage them. From this perspective, hand tracking is a must in our system and therefore different options are considered such as Leap Motion, Microsoft Kinect and OpenCv. Leap Motion [25] is more directed towards virtual reality applications, mainly mid-air grabbing. However, Leap Motion has a high error rate if the user rotates his hand quickly and its software development kit is not very flexible and doesn't allow much out of the set of provided functions. An alternative was Microsoft

[25] https://www.leapmotion.com

Kinect [26], however, Kinect SDK is not cross-platform and it is more complex than OpenCv. Therefore, for the simplicity of the implementation OpenCv was the suitable option. OpenCv (Open Source Computer Vision) is a free, cross-platform library aimed for providing functionality for real-time computer vision. The library is originally developed by Intel, and implemented in C and C++, however, it has interfaces for C++, Python and Java. Moreover, currently it supports the deep learning frameworks TensorFlow, Torch, PyTorch and Caffe, as most of the computer vision techniques are involving artificial intelligence. Nevertheless, the main advantage of the library is the enabling of multi-core processing and based on OpenCL [27] it has an advantage in the hardware acceleration of the underlying heterogeneous computing platform.

The user's hand tracking in our implementation is happening by means of tracking a coloured item (i.e red colour) over a 4×4 grid. The grid reflects to the 16 haptic motors in the device. Each X and Y position is then translated inside the communication script A.3.2 to an index between 1 and 16. Although, the original code for tracking used in our implementation is online [28], some modifications are performed. Firstly by reinforcing the code with a grid layer over the image and secondly by returning the position where the detection is happening based on the centroids of the detected coloured item. The coloured item will be placed on the user's hand. The tracking occurs constantly and as well does the position updates through the communication script, such that a haptic motor is continuously triggered as long as the user's hand is hovering over its position. However, if the user moved his hand to another position, the current motor will be stopped and the motor at the new position is triggered instead. Finally, if the user moved his hand away from the thermo-tactile surface, specifically away from the range of the grid, all the haptic motors will be off until new detection occurs.

5.3 Interface Design

After discussing the hardware and the software implementation of the thermo-tactile surface, it is important to describe the design perspective and the operational logic flow of the thermo-tactile surface. Starting with the design, the main idea is similar to the current practices of the design from TUI and data physicalisation. As the design pattern of the interface relatively complies to the *MCRpd* architecture illustrated in 2.2. The representation

[26] `http://bit.ly/MS_Kinect`
[27] `https://www.khronos.org/opencl/`
[28] `http://bit.ly/OpenCv_Code`

part exemplifies both, the physical representation sculpted in the *Thermo-Tactile* surface and the digital representation performed by the projection of the visualisation. However, in our design the physicalisation represented by the thermo-tactile surface is not always dependent on the visualisation. The thermo-tactile surface can behave as a stand-alone platform and is able to represent the data independently by its own without a need for additional visualisation support (e.g. video projection). The data representation in this case is considered as a *full representation* and it is contradictory to the MCRpd pattern, in which the physical representation has to have a complementary visualisation channel. Nevertheless, even when video projection is used in combination with the thermo-tactile surface, it is used to represent other dimensions of the data than the ones depicted by the surface. Therefore, the information that is represented by the physical artefact is not redundant in the visual part. Which means that the dataset information is shared on the system and each component holds a unique part of the dataset. In this way, one can evaluate the usability of the physicalisation (i.e the thermo-tactile) as stand-alone representation and also when it is combined with traditional visualisations.

Building the thermo-tactile surface as a grid (i.e. matrix) of a sixteen Peltier element and a vibrator, each has drawbacks to technologies barrier and to facilitate the prototyping process rather than a scientific interpretation. Mainly the limitation is about the total area occupied by the surface which is exponentially proportional to the number of elements used to construct the surface. The Peltier element is relatively large to be used in a user interface, as each element is 4×4×3.4mm. Therefore, combining more elements in one surface will increase the surface area exponentially. Also, space is needed around each Peltier element, as the heat will spread over the surface. Therefore, adjacent Peltier elements may interfere if no space is considered, resulting in an average heat value and consequently the user will not be able to distinguish the difference if existed. However, this spacing has a similar concept to the dot pitch between the screen's pixels and it can be of a smaller size if the technology advances in the future. Consequently, if smaller Peltier elements are made available, the dot pitch can be made smaller as well, which will result in a space gain. Also, one can increase the number of Peltiers in the surface's grid. Eventually, more data points with more complex ranges can be represented, as representing continuous data in the current prototype is challenging due to the small number of Peltier elements and their size. Although, the spacing is respected in our implementation, it has been recruited perfectly to embed the haptic motors. Such that each row in the grid is a mix of Peltiers and haptic motors. A row starts with a Peltier element and ends with a haptic motor, assuring that a haptic motor exist between each two adjacent Peltiers. Whereas inter-column spacing is left unoccupied in the current prototype, one may still consider this in a later iteration by, for example, increasing the number of haptic motors. Figure 5.21 gives an abstract view of the final distribution of the thermal and haptic parts wherein each Peltier corresponds to a Peltier element with a full thermal control logic and each V represents a vibrator with a full haptic control logic. The Peltier elements and the haptic motors are placed on

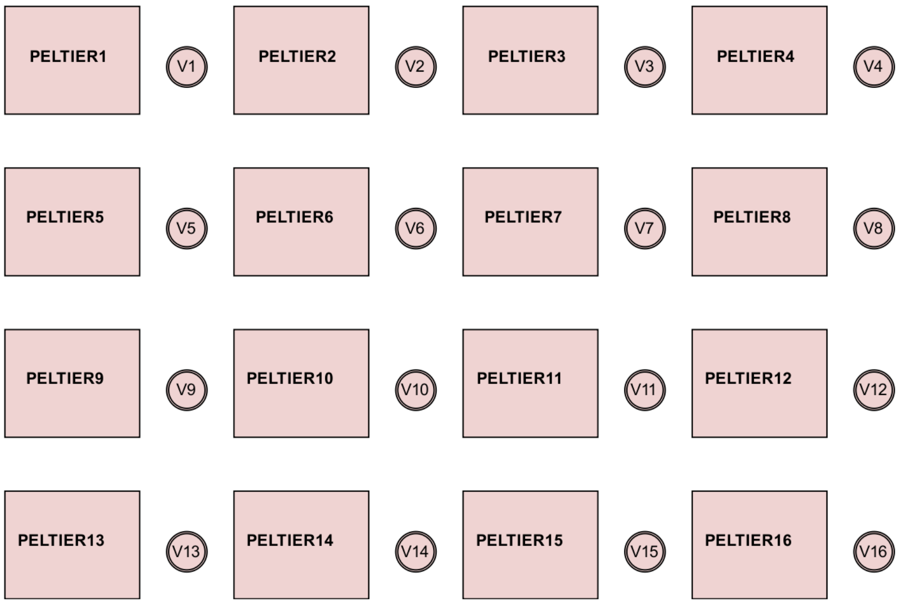


Figure 5.21: Thermal and haptic distribution

top of a wooden plate of size 25×25×1cm for the length, width and height respectively. Moreover, the electronic circuits of the items (i.e. Peltiers and Motors) are placed under the wooden plate and the wires pass through the plate. Then, all signal control cables are carefully attached to their predefined pins on the corresponding microcontroller. Thereafter, a copper plate is tightly placed on top of the items covering the whole space, such that the final prototype of the surface forms a tabletop-like shape, where a view is on the surface (i.e. screen) and the logic inside the table. Furthermore, each microcontroller passes a USB cable to the same computer and they are powered by that cable. Whereas, each part of the system is supplied by a different voltage source, such as 5V for the haptic part and 12V for the thermal part. After plugging the USB cables to the computer and the power supply cables to an electrical socket, the thermo-tactile surface is now operational.

As the thermo-tactile element is made of two parts, it has also two flows operating simultaneously. Both flows start by checking if the data exist on the internal memory (i.e hard-coded), otherwise they will read the data points sequentially from the computer over the serial connection and store them in the internal memory. After that, the thermal flow will keep operating only inside the microcontroller, unless a hard reset happens (i.e. manually clicking the reset button or turning it off then on). This hard reset can be used to update the data points, as the flow will start from the beginning. The microcontroller in turn will handle all the necessary configurations discussed earlier and starts generating the control signals to render the data on the Peltier. The Peltier element in its turn takes around 3 to 5 minutes rendering time to reach a steady state reflecting a data point to heat degree. Figure 5.22 shows the flow for one Peltier element. On the other hand and simultaneously

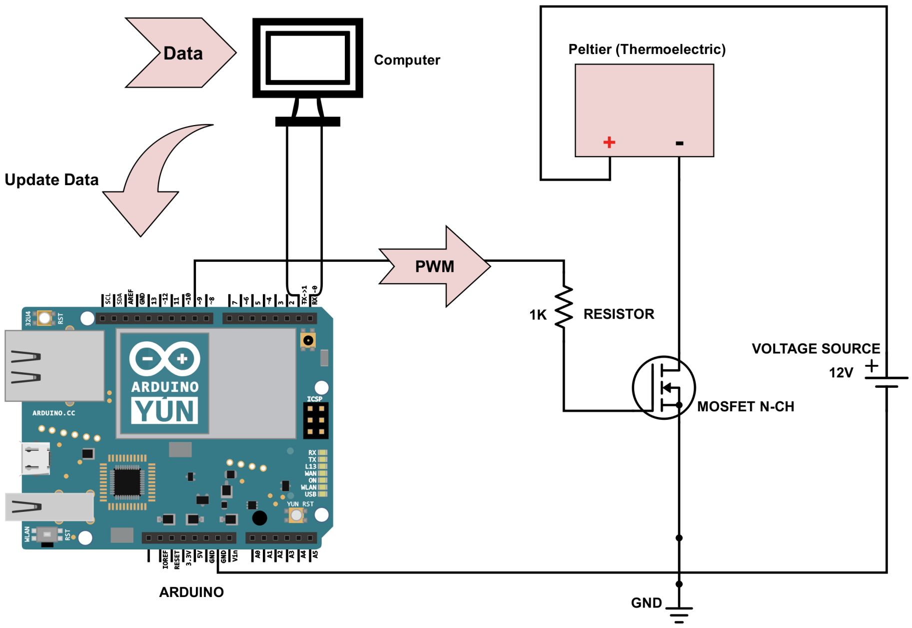


Figure 5.22: Thermal control flow

to the thermal flow, the haptic flow will continue inside the microcontroller which will handle all the needed configurations to setup the drivers and the waveforms to be triggered later. Thereafter, the microcontroller will return the I^{2}C bus to the ideal state and starts monitoring the serial port again, waiting for informations coming from the computer. On the other hand, the computer is running the hand tracking script to get position updates of the users hand from a webcam connected through a USB-port. Each position update is passed through the computer to the microcontroller over the serial port. The microcontroller in turn will handle the received position as an index to a haptic driver and decides if action has to be taken or not. Eventually, if an action is required, the microcontroller sends an I^2C signal to the corresponding index for triggering or braking. Constantly, the flow repeats itself throughout the running time. Finally, figure 5.23 shows the flow for four haptic drivers forming one row on the surface and Figure 5.24 represents the general view of the thermo-tactile surface, running as one coherent system .

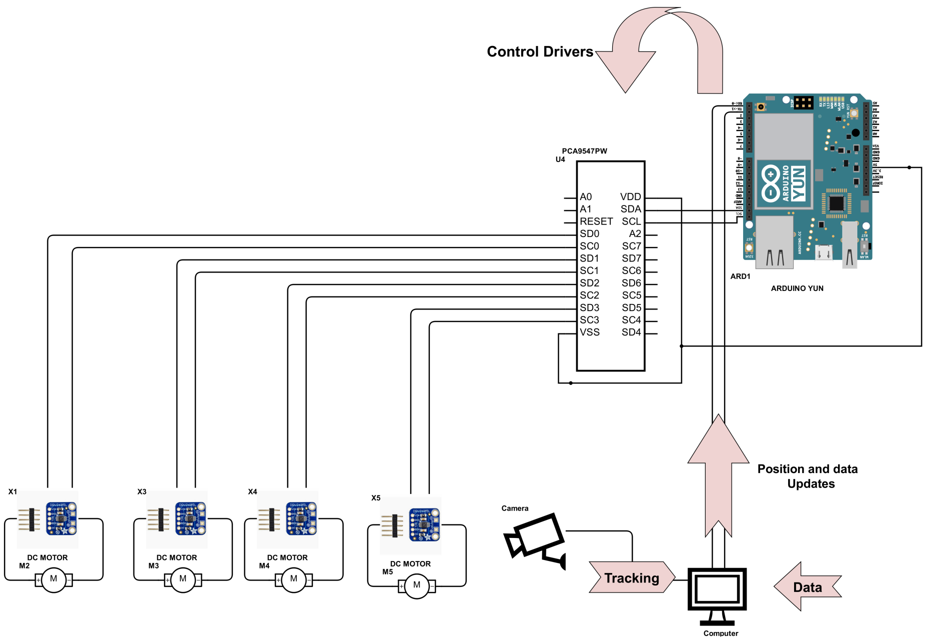


Figure 5.23: Haptic control flow

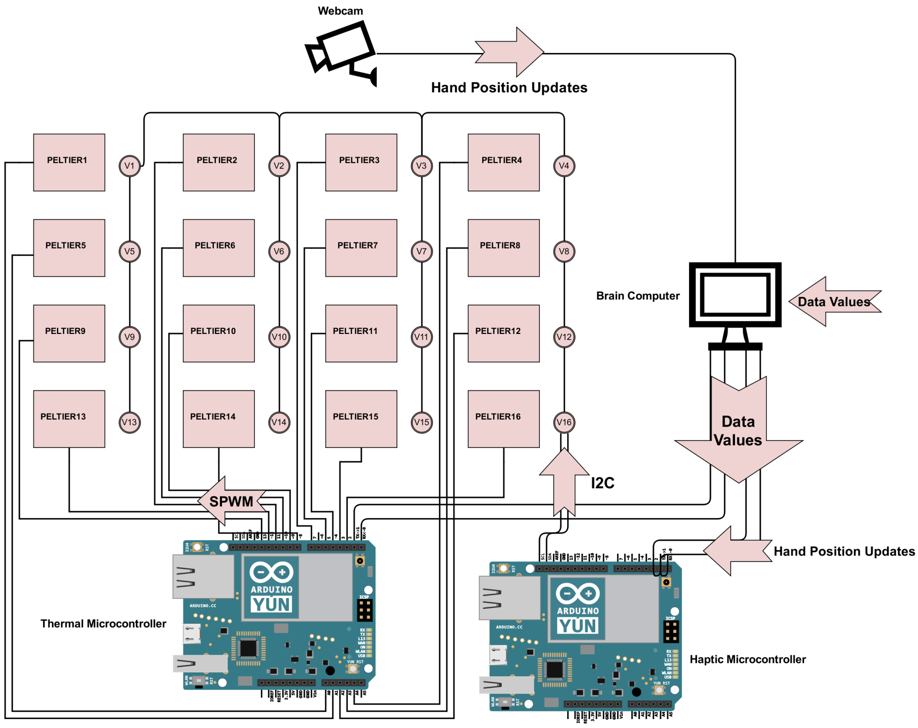
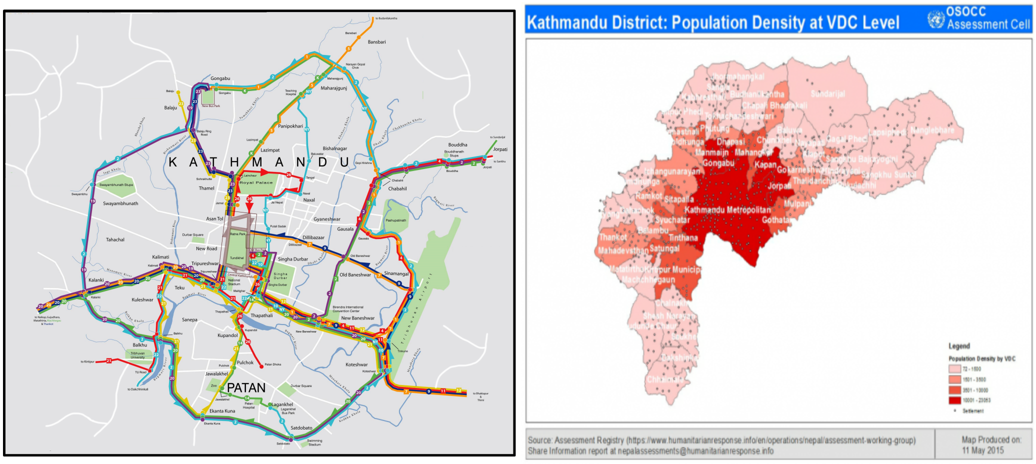


Figure 5.24: The general view of the thermo-tactile surface

6
The Experiment

An experiment is a scientific procedure undertaken to make a discovery, test and validate a hypothesis, or demonstrate a known fact. The goal of data physicalisation is to help people explore, understand, and communicate data [55]. Thus, there is a hypothesis that data physicalisations can provide effective representations of data which can be easily understood and communicated by users. Therefore, one needs to experiment and evaluate these physical representations in order to validate their effectiveness. However, it is still not clear what are the suitable evaluation methods for evaluating physicalisations [55]. Therefore, one of our research objectives is to investigate evaluating the effectiveness of physicalisations. Moreover, we developed the thermo-tactile interface as a prototype that employs different parameters from our proposed model, as well as to evaluate the effectiveness of physicalisation through haptic and thermal feedbacks. Unfortunately, in this thesis, we could not perform a full experiment due to the lack of time. However, we perform a lab study with four participants in order to validate the functionality of the prototype, as well as to get a preliminary insight into the effectiveness of physicalisation. The small test that we performed is not considered as a valid argument for the evaluation, as we have not collected any statistical data. However, as an exploratory evaluation, this test helps us to develop a better understanding on the evaluation for future works, as well as noting some positive and negative aspects regarding the current prototype,



(a) Regions map

(b) Population density © OSOCC

Figure 6.1: Nepal capital (Kathmandu)

such that one can improve the prototype for future experiments.

6.1 Setup

In order to evaluate our prototype, we recruited data for one of the worst natural disasters of the 21st century. Specifically, we are using the data of the Gorkha earthquake [1] that happened in Nepal. The earthquake struck Nepal in the centre of the country at a close distance from the capital Kathmandu that depicted in Figure 6.1a. The earthquake affected the capital from the north-west side as depicted in Figure 6.2. The disaster was massive, and the rescue teams were few in comparison to the impact of the earthquake. Therefore, a decision had to be taken to prioritise the post-disaster range and send the rescue teams to evacuate survivors in the zones that are most affected. The decision in such a situation should be taken very fast. The rational way of thinking is to evacuate the zones with high population density. The population density of the capital as depicted in Figure 6.1b, shows that the density is at maximum in the centre of the city and gets lower at the outskirts. In the same time, one has to consider the impact and the range of the earthquake because it can be that the earthquake fewer impacts the high-density areas. Thus it is clear from the description of the scenario that such a problem is a multi-criteria problem where different aspects are involved in forming the decision.

[1] `https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake`

Figure 6.2: Earthquake range map

In order to fit the visualisations with our prototype, we blended the impact of the earthquake directly on the map of the city, as depicted in Figure 6.3a. Similarly, the population density is converted to an array of boxes and blended on the map of the city. The boxes define regions in the city, and the choice of boxes is just for the simplicity. Finally, all the criteria are blended to represent the whole problem; this final visualisation is shown in Figure 6.3c, is the one being presented to the participants in the experiment. However, Figure 6.3c shows one of the limitations of visualisations in representing different heat-maps together, as some aspects of the data is getting blocked by other aspects.

After introducing the problem to the participants, the visualisation of the problem in Figure 6.3c is also explained to the participants. After that, the participants were asked to perform a simple task to prioritise the regions that should be evacuated first, based on the virtualisation of the problem. After they have done the first task, the participants were asked to perform the same task again but by exploring the problem on the thermo-tactile interface. The population values were represented by heat, whereas, vibration represented the earthquake intensity. Finally, the participants were asked to answer two question about their experience in using the thermo-tactile interface; these are listed below:

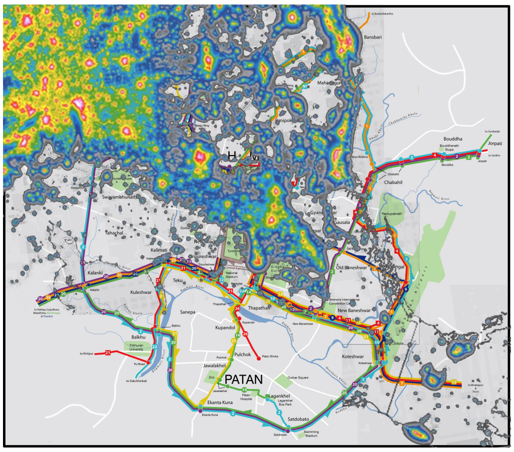
- **Address your positive observations about your experience in using the thermo-tactile interface?**

- **Address your negative observations about your experience in using the thermo-tactile interface?**

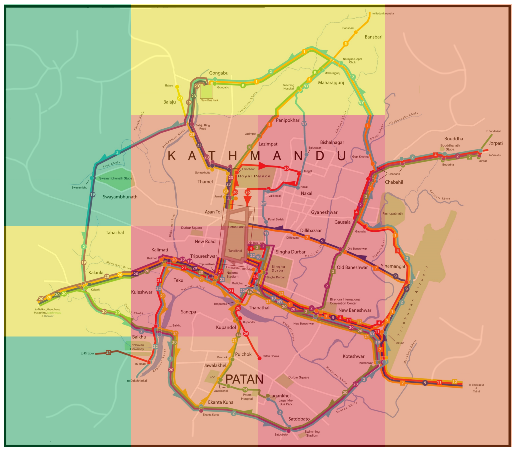
6.2 Results

Our exploratory evaluation took two hours to be accomplished. Each participant spent around thirty minutes in exploring and performing the tasks, as well as answering the two questions. According to our results, none of the participants could identify the highest priority regions when the explored the problem through the visualisation in Figure 6.3c. All participants referred to the top-left corner as the most important region. The participants failed in this task because they missed the information of the population in that corner since the earthquake impacts are blocking the population information in the representation. On the other hand, the participant performed better when the explored the task through the thermo-tactile interface. One participant referred to the centre exactly as the most important region. Whereas, the other three participants were close to the centre. The reason that participants performed better when using the thermo-tactile interface is that the pieces of information are not blocked by each other since each aspect is being encoded in an independent channel.

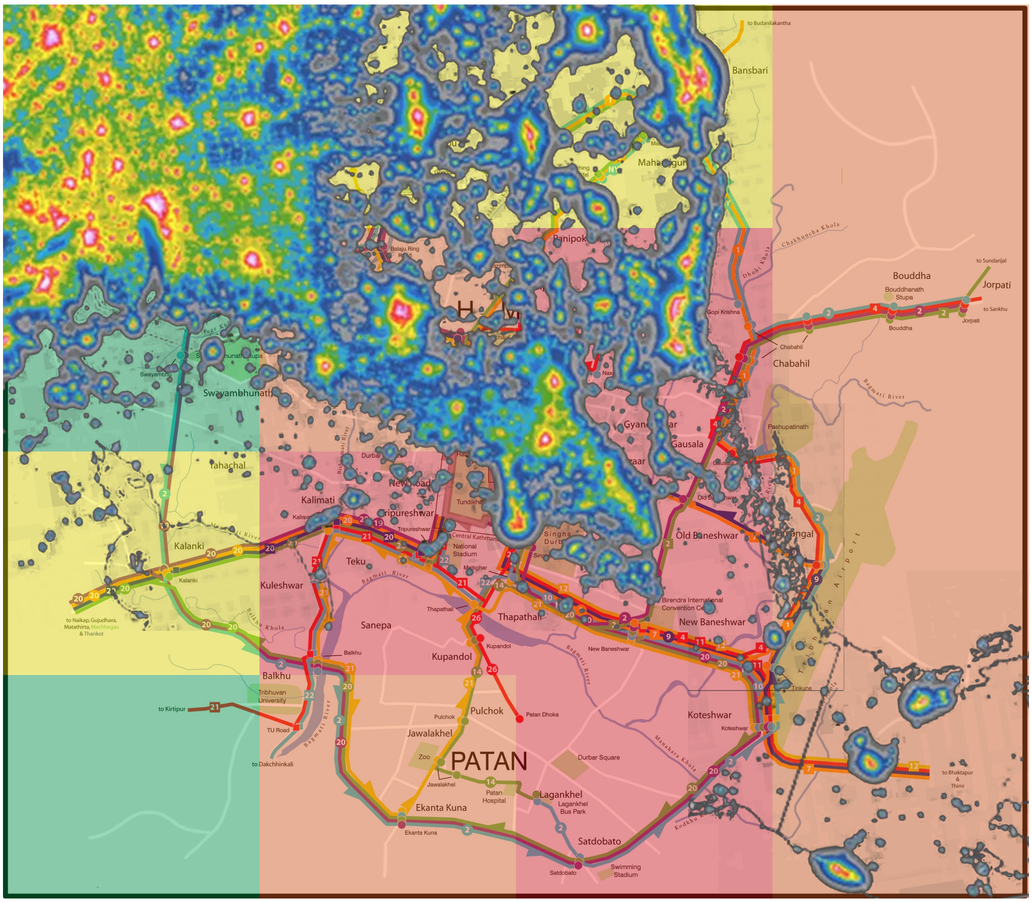
Regarding the two questions, the participants in the first question agreed that exploring the data through the thermo-tactile interface was a joyful experience and that they liked the idea of exploring data without looking at visualisations. Also, some participants mentioned the simplicity of using the interface and one thinks that the interface is nicely designed. In the second question, two participants noted that they could not correctly distinguish the heat difference between the centre and its surroundings. One participant mentioned that the inability to distinguish the difference of heat happened after some time but not at from the beginning of the exploration. This observation is explainable as human hands may get saturated with high heat values, such that low heat values will become imperceivable. Nevertheless, all participants agreed that the noise generated by the vibration on the metal is not pleasant. Finally, two participants suggested some improvements for the hand-tracking, as the vibrators may confuse the users by running when the hand is hovering over them.



(a) Earthquake impacts on the city



(b) Population density



(c) Final visualisation

Figure 6.3: The visualisations of data

7 Future Work and Conclusion

In this chapter, we highlight some aspects in the design of data physicalisation that merit further refinement and development in future works. After that we give some conclusions regarding the work undertaken in this thesis.

7.1 Future work

When designing an interactive systems, designers usually try to achieve the usability goals [91] in their design, such that the system should be effective to use, easy to learn and provide enjoyable user experience. Therefore, this interactive system goes through different iterations of design, starting with a requirement analysis and proceeding with design alternatives and then prototyping. Eventually, each new prototype will be evaluated at the end of each iteration. The evaluation will result in either starting a new iteration to refine the prototype, or a final product is concluded if the prototype achieved the usability goals and user's satisfaction is acquired. Therefore, the evaluation of the system is crucial to judge whether it is effective, efficient, easy to learn, easy to use, memorable, etc. Similarly, evaluating physicalisations is an important process to validate their usability. However, it is still not clear what are the the suitable evaluation methods for evaluating physicalisations [55]. Therefore, one of our research objectives was to evaluate the effectiveness of physicalisations. Unfortunately in this thesis we could not perform a proper experiment due to the lack of time. However, we have developed the thermo-tactile interface for the purpose of the evaluation. Therefore, in the future we are planning to perform a complete experiment to evaluate the effectiveness of physicalisation through haptic and thermal feedbacks. The planed experiment will evaluate the variance in an exploratory data analysis task and by using an ANOVA test. Also, there are two planned scenarios for the experiment. In the first scenario, the user will have to explore the data presented by the thermo-tactile interface without seeing the visualisation of this data, then the data will be presented to the user visually and they have to answer small questionnaire in order to elaborate their understanding in both cases. In the second scenario, the data will be shared between the thermo-tactile interface and visualisation projected on top of the interface. However, each parameter will be encoded with a unique dimension from the dataset, such that the data presented by thermal feedback is different than the data presented by the haptic feedback, eventually the projected visualisation is also different than both. In this case the whole system (i.e. thermo-tactile and visualisation) serves as one platform to represent multidimensional data (i.e more than three dimensions).

The design space of data physicalisation is complex [42, 55] and several aspects still need to be investigated in more detail. In this thesis, we have presented an elementary model for the design space of data physicalisation. However, the analysis of systems, variables and parameters must be investigated in depth in order to generalise the model of the design space and to understand the suitable transformations from the information domain to the physicalisation domain, like in Bertin's work [6]. Also, the relations between systems, variables, parameters and data, need to be investigated mathematically, as these mathematical representation can help in understanding the mappings between different modalities. Moreover, the magnitude of each parameter have to be investigated to understand the number of data points or dimension that can be encoded in that parameter. Moreover, in traditional interaction design, the CARE properties (Complementarity, Assignment, Redundancy and Equivalence), are recruited to characterise and assess the aspects of interaction in terms of the combination of modalities at the user level [22]. One may have to investigate similar properties in data physicalisation to facilitate the task to the designers when designing a multimodal system. Finally, Boy et al. [11] describe *visualisation literacy* as "*the ability to use well established data visualizations (e.g. line graphs) to handle information in an effective, efficient, and confident manner*". Therefore, one may need to define data *physicalisation literacy* similar to visualisation literacy, which in turn will open new applications for data physicalisation.

7.2 Conclusion

Data physicalisation is an emerging field of research that aims to facilitate users' tasks of exploration, analysis and interaction with data. Moreover, data physicalisation has some relations to the field of information visualisation, as both concern data representation tasks. Also, the interaction with physicalisations is related to the field of human-computer interactions. Although the field of data physicalisation is new, it has a promising future. Data physicalisation provides many advantages, such as providing rich, compelling and intuitive interactions to the users by utilising users' natural perceptual exploration skills. However, it has also some challenges regarding making data physical, enabling the interactions with physicalisations and evaluating physicalisations. The work undertaken this thesis highlighted some current limitations with the design space of data physicalisation, such as identifying the physical variables that can construct the physicalisation. Therefore, we contribute to the research towards solving this problem by performing a comprehensive survey in data physicalisation and its related fields. In our survey, we used the thematic analysis in order to find common patterns within the existing literature. Based one the results of the survey, we propose a classification model in form of modalities, parameters and forms. However, multisensory representations or physicalisations are mostly a combination of different modalities, which has been also materialised in different examples. Therefore, we introduced an analysis model for physicalisations and multisensory representations in term of systems. Also, our model suggests how one might begin to realise improvements in understanding the design space of data physicalisation. Therefore, we provide a mathematical representation for the relations between data, parameters, variables and systems. Systems are a function of variables, and variables are a function of parameters. Eventually, parameters are functions over data, where each parameter is a function over a data point. By representing physicalisation through mathematics one can get a better understanding over the design space physicalisation. However, these mathematical representations should be investigated in depth in order to come up with a general model.

The evaluation of physicalisation is still another challenge to be considered. Therefore, we developed a thermo-tactile interface that utilities haptic and thermal parameters to generate a physical representation. Based on our experience during the development, the existing technologies bring extra difficulty in designing physicalisations. For example, when we designed the thermo-tactile interface, we were limited by the size of the elements that are used to construct the interface, as well as their response time. These limitations are validated by our preliminary experiment. Users mostly complained about the noise of vibration and the granularity level of heat. However, they still admitted that the experience was joyful and that they could get insights about the difference between data points without looking to the visualisation. However, these results cannot be taken into account since there has not been any evaluation to these results. Nevertheless, the purpose of this experiment was to test the functionality of the surface in order to improve it for later experiments. Also, the advances in future technology may tackle the obstacles in current technologies, as smaller, smarter and faster technologies may appear. Consequently, one can adapt new technologies to design and implement better prototypes.

A
Appendix

A.1 Thermal Code

```
#include <Arduino.h>
// macros for precompile (Pin_state)
#define ON true
#define OFF false
// variables for software PWM (timing)
unsigned long currentMicros = micros();
unsigned long previousMicros = 0;
// variable for the frequency
// frequency = 1/(2 * microInterval)
// Note: Here is the frequency 10KHz
unsigned long microInterval = 100;
const int pwmMax = 255;
// typedef for properties of each sw pwm pin
typedef struct pwmPins
{
  int pin;
  unsigned int pwmOnValue;
  int pwmOffValue;
  bool pinState;
} pwmPin;
// variable to count the ticks of the pins (pwm_on)
unsigned int pwmTickCount;
//assgin the number of pins
```

```
const int pinCount = 16;
//creat the pins array
const byte arduinoPins[pinCount] = {2, 3, 4, 5, 6, 7, 8, 9, 10,
    11, 12, 13, 44, 45, 46, 47};
//Create the array for the data
int dataSetValues[pinCount];
//Set the range of the dataset distribution
int range_lower;
int range_upper;
//declare an array to store the values to be randered
int heatValues[pinCount];

//create a pwmPins type array
pwmPin softPWMpins[pinCount];

/*This function is responsible on mapping the dataset values to
    the range of PWM signal*/
void setupHeatValues(int lower, int upper)
{
  int valueHeat;
  int dataValue;
  for (int i = 0; i < pinCount; ++i)
  {
    dataValue = dataSetValues[i];
    valueHeat = map(dataValue, lower, upper, 0, 255);
    heatValues[i] = valueHeat;
  }

}
void setupPWM()
{
  int value;
  int valueHeat;
  //set the pwm_counter to zero as initialisation.
  pwmTickCount = 0;
  //loop over all pins
  for (int i = 0; i < pinCount; ++i)
  {
    //get the value of each pin from the dataset
    value = heatValues[i];
    //map it to the acceptable range (40 as maximum generates
    around 2v  (60 clicus) )
    valueHeat = map(value, 0, 255, 0, 40);
    //set the pin number in the softPWM array from the pins
    array
    softPWMpins[i].pin = arduinoPins[i];
    // the mapped value is actually the duty cycle (the amount
    of the pulse being on)
    softPWMpins[i].pwmOnValue = valueHeat;
```

```
    // the reset is the pwm off (the full pwm - time on)
    softPWMpins[i].pwmOffValue = pwmMax - valueHeat;
    // estiblish the pin as ON (true/HIGH) by default (as it is
   the off with the MOSFET NPN)
    softPWMpins[i].pinState = ON;
    //Normal setup for the pins
    pinMode(arduinoPins[i], OUTPUT);
  }
}
void runSPWM()
{
  currentMicros = micros();
  //check if the cycle his finished (to increment
   PWM_ticks_counter)
  if (currentMicros - previousMicros >= microInterval)
  {
    //Set, the previous time to the current for the next cycle
    previousMicros = currentMicros;
    //loop over all the pins to check their cycle
    for (int i = 0; i < pinCount; ++i)
    {
      //set the state to ON as default
      softPWMpins[i].pinState = ON;
      //if the number of ticks reachs the needed pwm on
      //(evalute to true after several runs, when the specific
   pin has already made the amount of pwm_on
      // If true, then change the state of the pin to OFF
      if (pwmTickCount >= softPWMpins[i].pwmOnValue)
      {
        softPWMpins[i].pinState = OFF;
      }
      //Check if the tick counter completed already a full pwm
   cycle
      //If true, switch the state back to ON and reset the
   counter
      if (pwmTickCount >= pwmMax)
      {
        // softPWMpins[i].pinState = ON;
        pwmTickCount = 0;
      }
      // Write to the pin based on the state (On = High, Off =
   Low)
      digitalWrite(softPWMpins[i].pin, softPWMpins[i].pinState);
    }
    //increment the tick counter to start new tick.
    pwmTickCount++;
  }
}
```

```
void setup()
{
  //Variabes to read the data from serial
  String data_range[2];
  String data_point;
  int read_counter = 0;
    /*Start Serial connection with the computer
  For communicating with the computer, use one of these rates:
  300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400,
    57600, or 115200.*/
  Serial.begin(115200);
    // Serial.setTimeout(500);
  Serial.println("Thermal Setup started");
  Serial.println("");
  /*Read from the serial to set the range and the data points*/
  if (Serial.available() > 0)
  {
    //Read the buffer untill the stop char (#)
    data_range[0] = Serial.readStringUntil('#');
    data_range[1] = Serial.readStringUntil('#');
    //Parse the string value to integer
    range_lower = data_range[0].toInt();
    range_upper = data_range[0].toInt();
    //Read the other values of the data
    while(read_counter < pinCount){
      data_point = Serial.readStringUntil('#');
      dataSetValues[read_counter] = data_point.toInt();
      read_counter++;
    }
  }
  setupHeatValues(range_lower, range_upper);
  setupPWM();
}

void loop()
{
  //Continous control.
  runSPWM();
}
```

Listing A.1: Thermal Control Logic

A.2 Haptic Code

A.2.1 DRV2605 Library

Header File

```
/*********************************************
```

```
  This is a library for the Adafruit DRV2605L Haptic Driver

  ----> http://www.adafruit.com/products/2305

  The Original code is written by Limor Fried/Ladyada for
    Adafruit Industries.

  This updated version is written by Ahmed Kareem A Abdullah.

  MIT license, all text above must be included in any
    redistribution
 ****************************************************/


#if ARDUINO >= 100
 #include "Arduino.h"
#else
 #include "WProgram.h"
#endif

#include <Wire.h>

#define DRV2605_ADDR 0x5A

#define DRV2605_REG_STATUS 0x00
#define DRV2605_REG_MODE 0x01
#define DRV2605_MODE_INTTRIG  0x00
#define DRV2605_MODE_EXTTRIGEDGE  0x01
#define DRV2605_MODE_EXTTRIGLVL  0x02
#define DRV2605_MODE_PWMANALOG  0x03
#define DRV2605_MODE_AUDIOVIBE  0x04
#define DRV2605_MODE_REALTIME  0x05
#define DRV2605_MODE_DIAGNOS  0x06
#define DRV2605_MODE_AUTOCAL  0x07


#define DRV2605_REG_RTPIN 0x02
#define DRV2605_REG_LIBRARY 0x03
#define DRV2605_REG_WAVESEQ1 0x04
#define DRV2605_REG_WAVESEQ2 0x05
#define DRV2605_REG_WAVESEQ3 0x06
#define DRV2605_REG_WAVESEQ4 0x07
#define DRV2605_REG_WAVESEQ5 0x08
#define DRV2605_REG_WAVESEQ6 0x09
#define DRV2605_REG_WAVESEQ7 0x0A
#define DRV2605_REG_WAVESEQ8 0x0B

#define DRV2605_REG_GO 0x0C
#define DRV2605_REG_OVERDRIVE 0x0D
```

```
#define DRV2605_REG_SUSTAINPOS 0x0E
#define DRV2605_REG_SUSTAINNEG 0x0F
#define DRV2605_REG_BREAK 0x10
#define DRV2605_REG_AUDIOCTRL 0x11
#define DRV2605_REG_AUDIOLVL 0x12
#define DRV2605_REG_AUDIOMAX 0x13
#define DRV2605_REG_RATEDV 0x16
#define DRV2605_REG_CLAMPV 0x17
#define DRV2605_REG_AUTOCALCOMP 0x18
#define DRV2605_REG_AUTOCALEMP 0x19
#define DRV2605_REG_FEEDBACK 0x1A
#define DRV2605_REG_CONTROL1 0x1B
#define DRV2605_REG_CONTROL2 0x1C
#define DRV2605_REG_CONTROL3 0x1D
#define DRV2605_REG_CONTROL4 0x1E
#define DRV2605_REG_VBAT 0x21
#define DRV2605_REG_LRARESON 0x22


class DRV2605 {
 public:

  DRV2605(void);
  boolean begin(void);

  void writeRegister8(uint8_t reg, uint8_t val);
  uint8_t readRegister8(uint8_t reg);
  void setWaveform(uint8_t slot, uint8_t w);
  void selectLibrary(uint8_t lib);
  void go(void);
  void stop(void);
  void setMode(uint8_t mode);
  void setRealtimeValue(uint8_t rtp);
  // Select ERM (Eccentric Rotating Mass) or LRA (Linear
    Resonant Actuator) vibration motor
  // The default is ERM, which is more common
  void useERM();
  void useLRA();

 private:

};
```

Listing A.2: DRV2605 Library Header

Implementation File

```
/**************************************************
  This is a library for the Adafruit DRV2605L Haptic Driver

```

```
  ----> http://www.adafruit.com/products/2305

  The Original code is written by Limor Fried/Ladyada for
    Adafruit Industries.

  This updated version is written by Ahmed Kareem A Abdullah.

  MIT license, all text above must be included in any
    redistribution
 **************************************************/


#if ARDUINO >= 100
 #include "Arduino.h"
#else
 #include "WProgram.h"
#endif

#include <Wire.h>

#define DRV2605_ADDR 0x5A

#define DRV2605_REG_STATUS 0x00
#define DRV2605_REG_MODE 0x01
#define DRV2605_MODE_INTTRIG  0x00
#define DRV2605_MODE_EXTTRIGEDGE  0x01
#define DRV2605_MODE_EXTTRIGLVL  0x02
#define DRV2605_MODE_PWMANALOG  0x03
#define DRV2605_MODE_AUDIOVIBE  0x04
#define DRV2605_MODE_REALTIME  0x05
#define DRV2605_MODE_DIAGNOS  0x06
#define DRV2605_MODE_AUTOCAL  0x07


#define DRV2605_REG_RTPIN 0x02
#define DRV2605_REG_LIBRARY 0x03
#define DRV2605_REG_WAVESEQ1 0x04
#define DRV2605_REG_WAVESEQ2 0x05
#define DRV2605_REG_WAVESEQ3 0x06
#define DRV2605_REG_WAVESEQ4 0x07
#define DRV2605_REG_WAVESEQ5 0x08
#define DRV2605_REG_WAVESEQ6 0x09
#define DRV2605_REG_WAVESEQ7 0x0A
#define DRV2605_REG_WAVESEQ8 0x0B

#define DRV2605_REG_GO 0x0C
#define DRV2605_REG_OVERDRIVE 0x0D
#define DRV2605_REG_SUSTAINPOS 0x0E
#define DRV2605_REG_SUSTAINNEG 0x0F
```

```
#define DRV2605_REG_BREAK 0x10
#define DRV2605_REG_AUDIOCTRL 0x11
#define DRV2605_REG_AUDIOLVL 0x12
#define DRV2605_REG_AUDIOMAX 0x13
#define DRV2605_REG_RATEDV 0x16
#define DRV2605_REG_CLAMPV 0x17
#define DRV2605_REG_AUTOCALCOMP 0x18
#define DRV2605_REG_AUTOCALEMP 0x19
#define DRV2605_REG_FEEDBACK 0x1A
#define DRV2605_REG_CONTROL1 0x1B
#define DRV2605_REG_CONTROL2 0x1C
#define DRV2605_REG_CONTROL3 0x1D
#define DRV2605_REG_CONTROL4 0x1E
#define DRV2605_REG_VBAT 0x21
#define DRV2605_REG_LRARESON 0x22


class DRV2605 {
 public:

  DRV2605(void);
  boolean begin(void);

  void writeRegister8(uint8_t reg, uint8_t val);
  uint8_t readRegister8(uint8_t reg);
  void setWaveform(uint8_t slot, uint8_t w);
  void selectLibrary(uint8_t lib);
  void go(void);
  void stop(void);
  void setMode(uint8_t mode);
  void setRealtimeValue(uint8_t rtp);
  // Select ERM (Eccentric Rotating Mass) or LRA (Linear
    Resonant Actuator) vibration motor
  // The default is ERM, which is more common
  void useERM();
  void useLRA();

 private:

};
```

Listing A.3: DRV2605 Library Implementation

A.2.2 Haptic Control Logic

```
#include <Arduino.h>
#include <Wire.h>
#include "DRV2605.h"

```

```
// Define the multiplixers addresses
#define TCAADDR_A 0x70
#define TCAADDR_B 0x71

//The number of motors in the circut
const int motors_count = 16;

// Initiate an instance of the haptic driver
DRV2605 driver;

//define the set of effects to be used
//The general Effect to be used (Buzz).
uint8_t effect = 14;

//The haptic values
int vibrationValues[motors_count];

//The estimated wait time per level
//The has been counted based on the number of ticks per second
    that the arduino generates
long wait_delay = 10;

//Create a TypeOff for the motors
typedef struct vibrators
{
    uint8_t gate;
    long waitTime;
    int vibrationValue;
    bool check_on;
    DRV2605 driver;
} vibrator;

vibrator motors[motors_count];

//Adress temp store for the reset
uint8_t address;

// To select a specific gate for transmission
void selectGate(uint8_t i)
{
    /* the gates are zero index.
  However, starting from 1 to get rid of the inviald Serial.read
    ().
  Which returns zero if there was no read */

    i = i - 1;

    //Module the number % 8, to convert it between 0-7
    uint8_t shiftBits = i % 8;
```

```
    if (i > 15)
        return;

    else if (i < 8)
    {
        address = TCAADDR_A;
    }
    else if (i >= 8)
    {
        address = TCAADDR_B;
    }
    else
    {
        return;
    }

    Wire.beginTransmission(address);
    Wire.write(1 << shiftBits);
    Wire.endTransmission();
    delay(2);
}
//To close the multiplixer gate and go back to standby status
    after transmission
void resetGate(uint8_t addr)
{
    Wire.beginTransmission(addr);
    Wire.write(0); // no channel selected
    Wire.endTransmission();
    delay(2);
}

void setupDrv(DRV2605 drv)
{
    /*Start the connection to the haptic driver*/
    // vab.check_on =
    drv.begin();

    /*select LRA library (1-5 ERM, 6 LRA)
    In case of Internal effects 4 is good*/
    drv.selectLibrary(4);

    /*Select the mode of the driver
   Internal effects = DRV2605_MODE_INTTRIG
  PWM = DRV2605_MODE_PWMANALOG*/
    drv.setMode(DRV2605_MODE_INTTRIG);

    /*These setting are for PWM-type usage*/

    /*Selecting LRA as motor.
```

```
     4x BRAKE factor,
     Medium Loop Gain,
     1.8x ERM/20x LRA BEMF gain.*/

    drv.writeRegister8(DRV2605_REG_FEEDBACK, 0xB6);

    /*Selecting ERM type of motors*/

    // drv.writeRegister8(DRV2605_REG_FEEDBACK, 0x36);

    /*Select Bidirectional (Open/Closed loop),
    Set (Brake stabilizer, LRA sample time, Braking time, IDISS
   time) to default.*/

    drv.writeRegister8(DRV2605_REG_CONTROL2, 0xDA);

    /* setting the noise_threshold to default,
   selecting ERM open loop,
   Setting Supply_COMP, DATA_FORMAT and LRA_DRIVE to default,
   Setting the N_PWM_ANALOG to PWM and set LRA to open loop mode
    */

    drv.writeRegister8(DRV2605_REG_CONTROL3, 0xA1);

}

void setVibrationValue(uint8_t iter, DRV2605 drv)
{
    /*set the effect to play.
  The for loop is for the intensity of the vibration in this
   position*/
    for (uint8_t j = 0; j < iter; j++)
    {
        drv.setWaveform(j, effect); // Waveform add effect
    }
    drv.setWaveform(iter, 0); // Waveform end
}
// The vriable to store the selected gate from serial port.
void triger(DRV2605 drv)
{
    // play the effect!
    drv.go();
}
void stop(DRV2605 drv)
{
    // stop the effect!
    drv.stop();
}
```

```

void initialise()
{
    /*Reset the MUX to standby*/
    resetGate(TCAADDR_A);
    delay(1000);
    resetGate(TCAADDR_B);
    delay(1000);

    /* Initialise the haptic drivers */
    for (int i = 0; i < motors_count; ++i)
    {
        motors[i].gate = i + 1;
        motors[i].waitTime = wait_delay;
        motors[i].vibrationValue = vibrationValues[i];
        motors[i].driver = driver;

        Serial.println("Checking gate: " + (String)i);
        selectGate(motors[i].gate);
        // delay(1000);
        if (!motors[i].driver.begin())
        {
            /* There was a problem detecting the DRV2605L ... check your connections */
            Serial.println("Ooops, no Haptic at Gate: " + (String)i + " detected ... Check your wiring!");
            //Try to setup the driver again
            setupDrv(motors[i].driver);
            while (1)
                ;
        }
        setupDrv(motors[i].driver);
        // delay(1000);

        setVibrationValue(motors[i].vibrationValue, motors[i].driver);

        // delay(1000);

        Serial.println("Gate: " + (String)i + " Successed");

        resetGate(address);

        // delay(1000);

        Serial.flush();
    }
}
// Current data index based on user's hand position
```

```
uint8_t data_index;
String data_index_str;
// A vriable to store the updated position each time
uint8_t new_data_index = -1;
String new_data_index_str;
// This variable to handle the delay after each trigger
long wait_counter = 0;
//This variable to handle the number of times that vibrator
    triggers
int ticks_counter = 0;

void setup()
{
    /*Start Serial connection with the computer
  For communicating with the computer, use one of these rates:
  300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400,
    57600, or 115200.*/
    Serial.begin(115200);
    // Serial.setTimeout(500);
    Serial.println("DRV Setup started");
    Serial.println("");
    /*Read from the serial to set the values of the vibrations
    and the data points*/
    if (Serial.available() > 0)
    {
        int read_counter = 0;
        while(read_counter < pinCount){
            data_point = Serial.readStringUntil('#');
            vibrationValues[read_counter] = data_point.toInt();
            read_counter++;
        }
    }
    //initialise the haptic drivers
    initialise();
    Serial.println("setup done...");
}
void loop()
{
    // reset the variable's values each loop
    data_index = -1;
    wait_counter = 0;
    ticks_counter = 0;
    //Check if there is data in the serial buffer
    if (Serial.available() > 0)
    {
        // data_index = Serial.parseInt();
        //Read the data untill the stop char (#)
        data_index_str = Serial.readStringUntil('#');
        Serial.println(data_index_str);
```

```
        //Parse the data to integer
        data_index = data_index_str.toInt();
        //Set the new and current index to the same value.
        new_data_index = data_index;
        //If the data index is vaild (>=1)
        if (data_index >= 1)
        {
            Serial.println("Debug msg: data recieved");
            Serial.println(data_index, DEC);
            //Set connection to that index
            selectGate(data_index);
            //Check the number of triggers of that index
            while (ticks_counter < motors[data_index - 1].vibrationValue)
            {
                ticks_counter++;
                Serial.println(ticks_counter, DEC);
                triger(motors[data_index - 1].driver);
                //Handle the wait time between each trigger
                while (new_data_index == data_index && motors[data_index - 1].waitTime >= wait_counter)
                {
                    wait_counter++;
                    Serial.println(wait_counter, DEC);
                    // new_data_index = Serial.parseInt();
                    //Check if the user still exploring the same position
                    new_data_index_str = Serial.readStringUntil('#');
                    new_data_index = new_data_index_str.toInt();
                }
                wait_counter = 0;
            }
            /*This is additional wait if and only if:
             - a haptic motor completed a full triggering cycle (x ticks).
             - The user still exploring the same position*/
            if (data_index == new_data_index)
            {
                Serial.println("Debug msg: new_i == i");
                delay(500);
            }
            //Stop the vibration and close the connection
            stop(motors[data_index - 1]);
            resetGate(address);
        }
        Serial.flush();
    }
```

```
}
```

Listing A.4: Haptic Control Logic

A.3 Interactions

A.3.1 Colour Tracking

```
#!/usr/bin/env python
# Initial code is taken from : https://github.com/simondlevy/OpenCV-Python-Hacks/blob/master/greenball_tracker.py
# There are many examples availave for multi-color object detection. I have no particular reason for choosing the above code

from time import sleep
import cv2
import numpy as np
import communication as com
# For OpenCV2 image display
WINDOW_NAME = 'ColorTracker'
cap_region_x_begin=0.1 # start point/total width
cap_region_y_end=0.9  # start point/total width
threshold = 60  #  BINARY threshold
lineThickness = 2
xdiv = 4
ydiv = 4

def drawGrid(rect,image,gridcelWidth,gridcelHeight):
    #Draw the horizontal lines
    for i in range(xdiv+1):
        lineX=rect[0]+i*gridcelWidth
        cv2.line(image, (lineX, rect[1]), (lineX, rect[3]),
    (0,255,0), lineThickness)

    # Draw the vertical lines
    for i in range(ydiv+1):
        lineY=rect[1]+i*gridcelHeight
        cv2.line(image, (rect[0], lineY), (rect[2], lineY),
    (0,255,0), lineThickness)

def track(image):

    # Blur the image to reduce noise
    blur = cv2.GaussianBlur(image, (5,5),0)

    # Convert BGR to HSV
    hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)
```

```

    # Threshold the HSV image for only green colors
    # For the moment red is the chosen color
    cLowerBound=np.array([20,124,123],np.uint8)
    cUpperBound=np.array([180,255,255],np.uint8)


    # Threshold the HSV image to get only green colors
    mask = cv2.inRange(hsv, cLowerBound, cUpperBound)
    cv2.namedWindow("mask",cv2.WINDOW_NORMAL)
    cv2.resizeWindow("mask",800,600)
    cv2.imshow("mask",mask)

    # Blur the mask
    bmask = cv2.GaussianBlur(mask, (5,5),0)

    # Take the moments to get the centroid
    moments = cv2.moments(bmask)
    m00 = moments['m00']
    centroid_x, centroid_y = None, None
    if m00 != 0:
        centroid_x = int(moments['m10']/m00)
        centroid_y = int(moments['m01']/m00)

    # Assume no centroid
    centroid = (-10,-10)

    # Use centroid if it exists
    if centroid_x != None and centroid_y != None:

        centroid = (centroid_x, centroid_y)

        # Put black circle in at centroid in image
        cv2.circle(image, centroid, 10, (0,0,0))



    rect = [int(cap_region_x_begin * image.shape[1]),#x0
        image.shape[0]-int(cap_region_y_end * image.shape[0]),#y0
        image.shape[1]-int(cap_region_x_begin * image.shape[1]),#x1
        int(cap_region_y_end * image.shape[0])]#y1

    gridcelHeight=int((rect[3]-rect[1])/ydiv)
    gridcelWidth=int((rect[2]-rect[0])/xdiv)

    # Display full-color image\
    drawGrid(rect,image,gridcelWidth,gridcelHeight)
```

```
    cv2.namedWindow(WINDOW_NAME, cv2.WINDOW_NORMAL)
    cv2.resizeWindow(WINDOW_NAME,800,600)
    cv2.imshow(WINDOW_NAME, image)
    cellIndex=-1
    if centroid[0] < rect[2] and centroid[1] < rect[3] and centroid[0] > rect[0] and centroid[1] > rect[1]:
        relCentroid = (centroid[0] - rect[0],centroid[1]-rect[1])
        cellIndex = (int(relCentroid[1]/gridcelHeight),int(relCentroid[0]/gridcelWidth))

    # Force image display, setting centroid to None on ESC key input
    if cv2.waitKey(1) & 0xFF == 27:
        cellIndex = -10

    # Return coordinates of centroid
    return cellIndex

# Test with input from camera
if __name__ == '__main__':

    capture = cv2.VideoCapture(0)

    while capture.isOpened():

        ret, frame = capture.read()

        if ret:

            cellIndex=track(frame)

            if cellIndex ==-10:
                break
            elif cellIndex != -1:
                cellNum = (cellIndex[0] * xdiv) + cellIndex[1]
                print(cellIndex)
                com.sendToSerial(cellNum)
                com.readFromSerial()

            if cv2.waitKey(1) & 0xFF == 27:
                break

        else:

           print('Capture failed')
           break
```

Listing A.5: Colour Tracking OpenCV

A.3.2 Colour Tracking

```
from time import sleep
import serial
# Arduino Uno
# portNumber = '/dev/tty.wchusbserialfd120'
# Mega2560
portNumber = '/dev/tty.usbmodemFA131'
boudRate = 115200
def connectToSerial():
  # Establish the connection on a specific port
  ser = serial.Serial(portNumber, boudRate)
  sleep(2)
  return ser


#to make the connection
serialPort = connectToSerial()
#to check the state of the serial port
serialOn = serialPort.is_open

def sendToSerial(cellNum):
  global serialPort, serialOn#, currentCell, counter
  if serialOn:
    # bytestring = cellNum.to_bytes(1, 'little')
    cellNum = cellNum + 1
    print("Current Cell : %s"%(cellNum))
    # Convert the decimal number to char and encode it, then send it to the Arduino
    # serialPort.write(('%d'%(cellNum)).encode())
    serialPort.write((str(cellNum) + '#').encode())
    serialPort.flush()
    # sleep(1) # Delay for one second

  else:
    serialPort = connectToSerial()




def readFromSerial():
  global serialPort, serialOn
  if serialOn:
    while (serialPort.in_waiting > 0):
      print (serialPort.readline()) # Read the newest output from the Arduino
    serialPort.reset_input_buffer()

  else:
```

```
    serialPort = connectToSerial()

# If to send data through the terminal
if __name__ == '__main__':
  serialPort = connectToSerial()
  while (serialPort.is_open):
    readFromSerial()
    x = int(input("Input a number:> "))
    sendToSerial(x)
```

Listing A.6: Serial Port Handler

Bibliography

[1] Jason Alexander, Yvonne Jansen, Kasper Hornbæk, Johan Kildal, and Abhijit Karnik. Exploring the Challenges of Making Data Physical. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 2417–2420, April 2015.

[2] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. Snake Charmer: Physically Enabling Virtual Objects. In *Proceedings of the TEI 2016: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 218–226, February 2016.

[3] O. Bailey. *Embedded Systems: Desktop Integration*. Wordware Pub., June 2005.

[4] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. NormalTouch and TextureTouch: High-fidelity 3D Haptic Shape Rendering on Handheld Virtual Reality Controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728, October 2016.

[5] Hrvoje Benko, Andrew D. Wilson, and Ravin Balakrishnan. Sphere: Multi-touch Interactions on a Spherical Display. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, pages 77–86, October 2008.

[6] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps.* Esri Press, 2011.

[7] Lonni Besançon, Mehdi Ammi, and Tobias Isenberg. Pressure-Based Gain Factor Control for Mobile 3D Interaction Using Locally-Coupled Devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1831–1842, May 2017.

[8] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. Mouse, Tactile, and Tangible Input for 3D Manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4727–4740, May 2017.

[9] Andrea Lorena Aldana Blanco, Steffen Grautoff, and Thomas Hermann. CardioSounds: Real-time Auditory Assistance for Supporting Cardiac Diagnostic and Monitoring. In *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*, pages 45:1–45:4, London, UK, August 2017.

[10] Jens Bornschein, Denise Bornschein, and Gerhard Weber. Comparing Computer-Based Drawing Methods for Blind People with Real-Time Tactile Feedback. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 115:1–115:13, Montreal, Canada, April 2018.

[11] J. Boy, R. A. Rensink, E. Bertini, and J. Fekete. A Principled Way of Assessing Visualization Literacy. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1963–1972, December 2014.

[12] Richard E Boyatzis. *Transforming Qualitative Information: Thematic Analysis and Code Development*. Sage, April 1998.

[13] R.L. Boylestad and L. Nashelsky. *Electronic Devices and Circuit Theory*. Pearson Education, April 2013.

[14] Stephen Brewster, David McGookin, and Christopher Miller. Olfoto: Designing a Smell-based Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 653–662, January 2006.

[15] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., June 1983.

[16] Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. UltraHaptics: Multi-point Mid-air Haptic Feedback for Touch Surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 505–514, October 2013.

[17] H. Cha, H. Lee, J. Park, H. S. Kim, S. C. Chung, and S. Choi. Mid-air Tactile Display Using Indirect Laser Radiation for Contour-following Stimulation and Assessment of its Spatial Acuity. In *2017 IEEE World Haptics Conference (WHC)*, pages 136–141, June 2017.

[18] Biplab Chakraborty, Debajit Sarma, Manas Bhuyan, and Karl MacDorman. A Review of Constraints on Vision-based Gesture Recognition for Human-Computer Interaction. *IET Computer Vision*, 12(1):3–15, January 2018.

[19] X. Chen, J. Z. Self, L. House, and C. North. Be the Data: A New Approach for lmmersive Analytics. In *2016 Workshop on Immersive Analytics (IA)*, pages 32–37, March 2016.

[20] Inrak Choi, Heather Culbertson, Mark R. Miller, Alex Olwal, and Sean Follmer. Grabity: A Wearable Haptic Interface for Simulating Weight and Grasping in Virtual Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 119–130, October 2017.

[21] Inrak Choi and Sean Follmer. Wolverine: A Wearable Haptic Interface for Grasping in VR. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 117–119, October 2016.

[22] Joëlle Coutaz, Laurence Nigay, Daniel Salber, Ann Blandford, Jon May, and Richard M. Young. *Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties*, pages 115–120. Boston, USA, July 1995.

[23] Alexandra Covaci, Gheorghita Ghinea, Chang-Hsin Lin, Shu-Hsien Huang, and Ju-Ling Shih. Multisensory Games-based Learning - Lessons Learnt from Olfactory Enhancement of a Digital Board Game. *Multimedia Tools and Applications*, 77(16):21245—21263, January 2018.

[24] Thomas Crespel, Brett Ridel, Clara Rigaud, Anke M. Brock, and Patrick Reuter. Code the Globe: Interactive Content for Spherical Multi-touch Displays with Simple Webpages. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays*, pages 14:1–14:6, June 2017.

[25] Alberto De Campo. Toward a Data Sonification Design Space Map. In *Proceedings of the 13th International Conference on auditory display*, pages 342–347, June 2007.

[26] Alan Dix, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction.* Prentice-Hall, Inc., February 2003.

[27] Paul Dourish. *Where the Action is: the Foundations of Embodied Interaction.* MIT press, August 2004.

[28] Bruno Dumas, Denis Lalanne, and Sharon Oviatt. *Multimodal Interfaces: A Survey of Principles, Models and Frameworks*, pages 3–26. Berlin, Heidelberg, June 2009.

[29] George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 442–449, October 1995.

[30] Sean Follmer, Daniel Leithinger, Alex Olwal, Akimitsu Hogge, and Hiroshi Ishii. inFORM: Dynamic Physical Affordances and Constraints Through Shape and Object Actuation. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 417–426, October 2013.

[31] MM. Gay-Lussac and Arago. *Annales de Chimie et de Physique.* Chez Grochard, Libraire, December 2008.

[32] Mark Grimshaw and Tom Garner. The Use of Sound to Represent Data and Concepts As a Means to Engender Creative Thought: Some Thoughts on Implementation and a Research Agenda. In *Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound*, pages 9–15, Corfu, Greece, September 2012.

[33] Florian Grond and Thomas Hermann. Interactive Sonification for Data Exploration: How listening modes and display purposes define design guidelines. *Organised Sound*, 19(1):41–51, April 2014.

[34] Z. Halim, R. Baig, and S. Bashir. Sonification: a Novel Approach Towards Data Mining. In *2006 International Conference on Emerging Technologies*, pages 548–553, November 2006.

[35] Chris Harding, Ioannis A. Kakadiaris, John F. Casey, and R.Bowen Loftin. A Multi-sensory System for the Investigation of Geoscientific Data. *Computers & Graphics*, 26(2):259–269, April 2002.

[36] Thomas Hermann. Taxonomy and Definitions for Sonification and Auditory Display. In *Proceedings of the 14th International Conference on Auditory*, June 2008.

[37] Thomas Hermann, Till Bovermann, Eckard Riedenklau, and Helge Ritter. Tangible Computing for Interactive Sonification of Multivariate Data. In *Proceedings of the 2nd International Workshop on Interactive Sonification*, page Workshop, February 2007.

[38] Thomas Hermann and Helge Ritter. Listen to your Data: Model-Based Sonification for Data Analysis. In *Advances in Intelligent Computing and Multimedia Systems*, pages 189–194, June 1999.

[39] V. Himpe. *Mastering the I2C Bus*. Elektor International Media, August 2011.

[40] Trevor Hogan, Uta Hinrichs, and Eva Hornecker. The Visual and Beyond: Characterizing Experiences with Auditory, Haptic and Visual Data Representations. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 797–809, June 2017.

[41] Trevor Hogan and Eva Hornecker. In Touch with Space: Embodying Live Data for Tangible Interaction. In *TEI 2013 - Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pages 275–278, February 2013.

[42] Trevor Hogan and Eva Hornecker. Towards a Design Space for Multisensory Data Representation. *Interacting with Computers*, 29(2):147–167, May 2017.

[43] Tobias Höllerer, JoAnn Kuchera-Morin, and Xavier Amatriain. The allosphere: A large-scale immersive surround-view instrument. In *Proceedings of the 2007 Workshop on Emerging Displays Technologies: Images and Beyond: The Future of Displays and Interacton*, page 3, August 2007.

[44] Eva Hornecker and Jacob Buur. Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 437–446, Montreal, Canada, April 2006.

[45] Lode Hoste, Bruno Dumas, and Beat Signer. Mudra: A Unified Multimodal Interaction Framework. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, pages 97–104, Alicante, Spain, November 2011.

[46] Carmen Hull and Wesley Willett. Building with Data: Architectural Models As Inspiration for Data Physicalization. In *Proceedings of the*

2017 CHI Conference on Human Factors in Computing Systems, pages 1217–1264, October 2017.

[47] S. Inoue, Y. Makino, and H. Shinoda. Active Touch Perception Produced by Airborne Ultrasonic Haptic Hologram. In *2015 IEEE World Haptics Conference (WHC)*, pages 362–367, June 2015.

[48] Hiroshi Ishii, Dávid Lakatos, Leonardo Bonanni, and Jean-Baptiste Labrune. Radical Atoms: Beyond Tangible Bits, Toward Transformable Materials. *interactions*, 19(1):38–51, January 2012.

[49] Hiroshi Ishii, Daniel Leithinger, Sean Follmer, Amit Zoran, Philipp Schoessler, and Jared Counts. TRANSFORM: Embodiment of "Radical Atoms" at Milano Design Week. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 687–694, April 2015.

[50] Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 234–241, March 1997.

[51] Y. Jansen and P. Dragicevic. An Interaction Model for Visualizations Beyond The Desktop. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2396–2405, Dec 2013.

[52] Y. Jansen and K. Hornbæk. A Psychophysical Investigation of Size as a Physical Variable. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):479–488, January 2016.

[53] Yvonne Jansen. *Physical and Tangible Information Visualization.* PhD thesis, March 2014.

[54] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the Efficiency of Physical Visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2593–2602, October 2013.

[55] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. Opportunities and Challenges for Data Physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3227–3236, April 2015.

[56] Kenneth K. Choo Joseph C. Stevens. Temperature Sensitivity of the Body Surface over the Life Span. *Somatosensory & Motor Research*, 15(1):13–28, July 1998.

[57] Cindy Hsin-Liu Kao, Ermal Dreshaj, Judith Amores, Sang-won Leigh, Xavier Benavides, Pattie Maes, Ken Perlin, and Hiroshi Ishii. Clayodor: Retrieving Scents Through the Manipulation of Malleable Material. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 697–702, January 2015.

[58] H. G. Kaper, E. Wiebel, and S. Tipei. Data Sonification and Sound Visualization. *Computing in Science Engineering*, 1(4):48–58, July 1999.

[59] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E. Froehlich. MakerWear: A Tangible Approach to Interactive Wearable Creation for Children. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 133–145, May 2017.

[60] Boriana Koleva, Steve Benford, Kher Hui Ng, and Tom Rodden. A Framework for Tangible User Interfaces. In *In workshop Proc. on Real World User Interfaces, Mobile HCI Conference 2003*, pages 257–264, July 2003.

[61] Susan J. Lederman and Roberta L. Klatzky. Extracting Object Properties Through Haptic Exploration. *Acta Psychologica*, 84(1):29–40, June 1993.

[62] Daniel Leithinger, Sean Follmer, Alex Olwal, Samuel Luescher, Akimitsu Hogge, Jinha Lee, and Hiroshi Ishii. Sublimate: State-changing Virtual and Physical Rendering to Augment Interaction with Shape Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1441–1450, April 2013.

[63] Daniel Leithinger and Hiroshi Ishii. Relief: A Scalable Actuated Shape Display. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 221–222, January 2010.

[64] R. Bowen Loftin. Multisensory Perception: Beyond the Visual in Visualization. *Computing in Science and Engeneering*, 5(4):56–58, July 2003.

[65] T. M. Madhyastha and D. A. Reed. Data Sonification: Do You See What I Hear? *IEEE Software*, 12(2):45–56, March 1995.

[66] Thomas H. Massie. The PHANToM Haptic Interface: A Device for Probing Virtual Objects. In *295–301*, May 1994.

[67] Matthias Milczynski, Thomas Hermann, Till Bovermann, and Helge Ritter. A Malleable Device with Applications to Sonification-based Data Exploration. In *Proceedings of the International Conference on Auditory Display*, pages 69–76, July 2006.

[68] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, March 1956.

[69] Andrew Vande Moere. Towards Designing Persuasive Ambient Visualization. In *Issues in the Design & Evaluation of Ambient Information Systems Workshop*, pages 48–52, May 2007.

[70] Andrew Vande Moere. Beyond the Tyranny of the Pixel: Exploring the Physicality of Information Visualization. In *Proceedings of the 2008 12th International Conference Information Visualisation*, pages 469–474, Washington, DC, USA, July 2008.

[71] Martin Murer, Ilhan Aslan, and Manfred Tscheligi. LOLLio: Exploring Taste As Playful Modality. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pages 299–302, February 2013.

[72] N. Murray, B. Lee, Y. Qiao, and G. Miro-Muntean. The Impact of Scent Type on Olfaction-Enhanced Multimedia Quality of Experience. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(9):2503–2515, September 2017.

[73] Takuji Narumi, Shinya Nishizaka, Takashi Kajinami, Tomohiro Tanikawa, and Michitaka Hirose. Augmented Reality Flavors: Gustatory Display Based on Edible Marker and Cross-modal Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 93–102, May 2011.

[74] Keith V. Nesbitt. Modeling the Multi-sensory Design Space. In *Proceedings of the 2001 Asia-Pacific Symposium on Information Visualisation - Volume 9*, pages 27–36, July 2001.

[75] Keith V Nesbitt. A Framework to Support the Designers of Haptic, Visual and Auditory Displays. In *Proceedings of GOTHI-05 Guidelines On Tactile and Haptic Interactions*, pages 54–64, July 2005.

[76] Laurence Nigay and Joëlle Coutaz. A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems*, pages 172–178, April 1993.

[77] Marianna Obrist, Alexandre N. Tuch, and Kasper Hornbaek. Opportunities for Odor: Experiences with Smell and Implications for Technology. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pages 2843–2852, April 2014.

[78] Marianna Obrist, Carlos Velasco, Chi Thanh Vi, Nimesha Ranasinghe, Ali Israr, Adrian D. Cheok, Charles Spence, and Ponnampalam Gopalakrishnakone. Touch, Taste, and Smell User Interfaces: The Future of Multisensory HCI. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3285–3292, May 2016.

[79] Philipp Offermann, Olga Levina, Marten Schönherr, and Udo Bub. Outline of a Design Science Research Process. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, pages 7:1–7:11, May 2009.

[80] T. Ogi and M. Hirose. Multisensory Data Sensualization Based on Human Perception. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, pages 66–70, 266, March 1996.

[81] Sharon Oviatt. Ten Myths of Multimodal Interaction. *Communications of the ACM*, 42(11):74–81, November 1999.

[82] S. Paneels and J. C. Roberts. Review of Designs for Haptic Data Visualization. *IEEE Transactions on Haptics*, 3(2):119–137, April 2010.

[83] D. Paret and C. Fenger. *The I2C Bus: from Theory to Practice*. Wiley, February 1997.

[84] Ken Peffers, Marcus Rothenberger, Tuure Tuunanen, and Reza Vaezi. Design Science Research Evaluation. In *International Conference on Design Science Research in Information Systems*, pages 398–410, May 2012.

[85] Zachary Pousman and John Stasko. A Taxonomy of Ambient Information Systems: Four Patterns of Design. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 67–74, May 2006.

[86] Nimesha Ranasinghe, Pravar Jain, Shienny Karwita, David Tolley, and Ellen Yi-Luen Do. Ambiotherm: Enhancing Sense of Presence in Virtual Reality by Simulating Real-World Environmental Conditions. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1731–1742, May 2017.

[87] Nimesha Ranasinghe, Kasun Karunanayaka, Adrian David Cheok, Owen Noel Newton Fernando, Hideaki Nii, and Ponnampalam Gopalakrishnakone. Digital Taste and Smell Communication. In *Proceedings of the 6th International Conference on Body Area Networks*, pages 78–84, November 2011.

[88] Nimesha Ranasinghe, Kuan-Yi Lee, Gajan Suthokumar, and Ellen Yi-Luen Do. The Sensation of Taste in the Future of Immersive Media. In *Proceedings of the 2Nd ACM International Workshop on Immersive Media Experiences*, pages 7–12, November 2014.

[89] Nimesha Ranasinghe, Gajan Suthokumar, Kuan Yi Lee, and Ellen Yi-Luen Do. Digital Flavor Interface. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 47–48, October 2014.

[90] Jonathan C. Roberts and Rick Walker. Using All Our Senses: the need for a Unified Theoretical Approach to Multi-sensory Information Visualization. 2010.

[91] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, February 2011.

[92] Deepak Ranjan Sahoo, Takuto Nakamura, Asier Marzo, Themis Omirou, Michihiro Asakawa, and Sriram Subramanian. JOLED: A Mid-air Display Based on Electrostatic Rotation of Levitated Janus Objects. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, October 2016.

[93] Nur Amira Samshir, Nurafiqah Johari, Kasun Karunanayaka, and Adrian David Cheok. Thermal Sweet Taste Machine for Multisensory Internet. In *Proceedings of the Fourth International Conference on Human Agent Interaction*, pages 325–328, October 2016.

[94] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Mühlhäuser. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1001–1014, May 2017.

[95] Samuel B. Schorr and Allison M. Okamura. Fingertip Tactile Devices for Virtual Object Manipulation and Exploration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3115–3119, May 2017.

[96] Orit Shaer and Eva Hornecker. Tangible User Interfaces: Past, Present, and Future Directions. *Found. Trends Hum.-Comput. Interact.*, 3(1–2):1–137, January 2010.

[97] Beat Signer and Timothy J. Curtin. Tangible Holograms: Towards Mobile Physical Augmentation of Virtual Objects. *CoRR*, abs/1703.08288, March 2017.

[98] Alexa F. Siu, Eric J. Gonzalez, Shenli Yuan, Jason Ginsberg, Allen Zhao, and Sean Follmer. ShapeShift: A Mobile Tabletop Shape Display for Tangible and Haptic Interaction. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, pages 77–79, October 2017.

[99] Rajinder Sodhi, Ivan Poupyrev, Matthew Glisson, and Ali Israr. AIREAL: Interactive Tactile Experiences in Free Air. *ACM Trans. Graph.*, 32(4):134:1–134:10, July 2013.

[100] Daniel Spelmezan, Deepak Ranjan Sahoo, and Sriram Subramanian. Sparkle: Hover Feedback with Touchable Electric Arcs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3705–3717, May 2017.

[101] Charles Spence. Multisensory Flavor Perception. *Cell*, 161(1):24–35, March 2015.

[102] Martin Spindler and Raimund Dachselt. Exploring Information Spaces by Using Tangible Magic Lenses in a Tabletop Environment. In *CHI 2010 Extended Abstracts on Human Factors in Computing Systems*, pages 4771–4776, April 2010.

[103] A. A. Stanley, K. Hata, and A. M. Okamura. Closed-Loop Shape Control of a Haptic Jamming Deformable Surface. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2718–2724, May 2016.

[104] Paul Strohmeier and Kasper Hornbæk. Generating Haptic Textures with a Vibrotactile Actuator. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4994–5005, May 2017.

[105] Simon Stusak, Jeannette Schwarz, and Andreas Butz. Evaluating the Memorability of Physical Visualizations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3247–3250, April 2015.

[106] Faisal Taher, John Hardy, Abhijit Karnik, Christian Weichel, Yvonne Jansen, Kasper Hornbæk, and Jason Alexander. Exploring Interactions with Physically Dynamic Bar Charts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3237–3246, April 2015.

[107] Faisal Taher, Yvonne Jansen, Jonathan Woodruff, John Hardy, Kasper Hornbaek, and Jason Alexander. Investigating the Use of a Dynamic Physical Bar Chart for Data Exploration and Presentation. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):451–460, January 2017.

[108] Susanne Tak and Alexander Toet. Towards Interactive Multisensory Data Representations. In *GRAPP/IVAPP*, pages 558–561, February 2013.

[109] R. A. Taylor and G. L. Solbrekken. Comprehensive System-Level Optimization of Thermoelectric Devices for Electronic Cooling Applications. *IEEE Transactions on Components and Packaging Technologies*, 31(1):23–31, March 2008.

[110] Simon Sherratt Terence K. L. Towards Disappearing User Interfaces for Ubiquitous Computing: Human Enhancement from Sixth Sense to Super Senses. *Journal of Ambient Intelligence and Humanized Computing*, 8(3):449–465, June 2017.

[111] Jordan Tewell, Jon Bird, and George R. Buchanan. Heat-Nav: Using Temperature Changes As Navigation Cues. In *Proceedings of the 2017*

CHI Conference on Human Factors in Computing Systems, pages 1131–1135, May 2017.

[112] Martin Richardson Tove N. Dalenius, Simon Rees. Holographic Data Visualization: Using Synthetic Full-Parallax Holography to Share Information, April 2017.

[113] B. Ullmer and H. Ishii. Emerging Frameworks for Tangible User Interfaces. *IBM Syst. J.*, 39(3-4):915–931, July 2000.

[114] Brygg Ullmer and Hiroshi Ishii. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, pages 223–232, New York, USA, October 1997.

[115] John Underkoffler and Hiroshi Ishii. Urp: A Luminous-tangible Workbench for Urban Planning and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 386–393, Pittsburgh, USA, May 1999.

[116] Elise van den Hoven and Berry Eggen. *Tangible Computing in Everyday Life: Extending Current Frameworks for Tangible User Interfaces with Personal Objects*. Springer Berlin Heidelberg, 2004.

[117] Andrew Vande Moere and Stephanie Patel. The Physical Visualization of Information: Designing Data Sculptures in an Educational Context. *Journal of urban technology*, 19(2):1–23, October 2010.

[118] Braun Virginia and Clarke Victoria. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology*, 3(2):77–101, July 2006.

[119] Bruce N. Walker, Jeffrey Lindsay, Amanda Nance, Yoko Nakano, Dianne K. Palladino, Tilman Dingler, and Myounghoon Jeon. Spearcons (Speech-Based Earcons) Improve Navigation Performance in Advanced Auditory Menus. *Human Factors*, 55(1):157–182, July 2013.

[120] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 3095–3105, May 2017.

[121] Wataru Yamada, Kazuhiro Yamada, Hiroyuki Manabe, and Daizo Ikeda. iSphere: Self-Luminous Spherical Drone Display. In *Proceedings*

of the 30th Annual ACM Symposium on User Interface Software and Technology, pages 635–643, October 2017.

[122] Haixia Zhao. Interactive Sonification of Geo-referenced Data. In *CHI 2005 Extended Abstracts on Human Factors in Computing Systems*, pages 1134–1135. ACM, 2005.

[123] Haixia Zhao, Catherine Plaisant, Ben Shneiderman, and Ramani Duraiswami. Sonification of Geo-Referenced Data for Auditory Information Seeking: Design Principle and Pilot Study. In *ICAD*, January 2004.

[124] Jack Zhao and Andrew Vande Moere. Embodiment in Data Sculpture: A Model of the Physical Visualization of Information. In *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, pages 343–350, New York, USA, September 2008.