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

TANGHO - A TANGIBLE HOLOGRAM PLATFORM FOR EXPLORING INTERACTIONS WITH DATA THROUGH MOBILE SIMULATED DATA PHYSICALISATION

TIMOTHY JOSEPH CURTIN

Academic year 2016 – 2017

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

Abstract

The last couple of decades has seen a steady increase in the production of data, while a broadening scope of utility and a deepening social presence has accompanied its growth. Indeed, data has become an ever-present part of life today, even finding representation and expression outside of traditional forms, permeating into areas such as fashion and the arts, for example. Its increasing complexity and polymorphous nature demand for novel and effective approaches to facilitate its exploration and analysis.

In tandem with this growth in data, the same period has witnessed the emergence and evolution of Tangible User Interfaces (TUIs), systems that attempt to bridge the physical and digital divide, thereby affording more direct and engaging interactions with virtual content. The vision of Radical Atoms, proposed by Hiroshi Ishii and others, extends the horizon further by considering what might be possible with *programmable matter*, a dynamic reconfigurable future material. The vision has already inspired innovative technologies such as shape changing interfaces, which allows for dynamic bidirectional input/output.

The proliferation of data to its present state of ubiquity, the promise of programmable matter and current technologies working towards it, and the arrival of accessible fabrication technologies such as 3D printing, have led to a growing interest in the idea of *physicalising* data. The benefits of doing so are manifold and the field of data physicalisation, which seeks to encode data with physical variables, has emerged in recent years to fulfil this ambition. Engaging with physicalised data allows one to leverage natural spatial and haptic perception skills in their interactions, increasing the channels with which one can perceive and comprehend, and supports the exploration and analysis of data in an intuitive and memorable way.

Given the promise of programmable matter and the important role it is certain to play in the field of data physicalisation, one might naturally start to think about the possibility of emulating the experience of working with programmable matter through existing technology. In doing this, technologists and interaction designers could, in the spirit of vision-driven research, begin to design interactions with potential near-future materials and programmable matter—moving beyond the storyboard to direct engagement with the possibilities. New materials might be conceptualised, prototyped and evaluated for behaviour and affordances for interaction, before deciding whether they are worth actualising.

In this thesis we explore the idea of simulating programmable matter and investigate those concerns most central to the development of a system with this capacity. With due consideration of the results of this investigation, we put forward the tangible hologram system as a response to the question of how to best approximate the full potential of programmable matter, with this tangible hologram approach entailing the haptic augmentation of holographically manifested digital information and data. We then set out how this platform may help to address the implementation and interaction challenges facing those working within the field of data physicalisation. To assess the viability of realising the tangible hologram solution, we implement a prototype system comprising a custom wearable mobile haptic unit and a Microsoft HoloLens, which provides the holographic rendering. Finally, we present a number of use cases to demonstrate how the system may be used to generate rich and engaging interactions with simulated programmable matter.

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to Professor Dr. Beat Signer for his perpetual support and clear guidance throughout the last year. The professionalism he displayed in managing the project and his constant availability were of great benefit and this was much appreciated. His genuine enthusiasm for, and belief in, the tangible hologram system proved to be an enduring source of motivation and encouragement in challenging times. Working with Professor Signer on the tangible holograms project has been an enriching and engaging experience and I hope that the TangHo system we started together can continue to grow and develop, fulfilling all the promise it holds.

I would also like to acknowledge the incredible support I received from my fiancée Eileen and my family back home in Ireland. Eileen's unwavering belief in me over the last year kept me going and will always be remembered.

Finally, I would like to thank the members of the WISE lab for their support and kindness throughout the year. I would especially like to thank Payam for his encouragement and welcome advice. I wish everyone in the lab great success in all their endeavours.

Contents

1 Introduction

The presented tangible hologram (TangHo) system is a platform that combines *mixed reality* and haptic technologies to augment the holograms generated by mixed reality with physical properties, thereby rendering them more real and tangible to a user. Mixed reality headsets such as the Microsoft HoloLens[1] and Meta2[2] have opened up a new world of possibilities when it comes to interacting with digital information. Through their ability to situate virtual objects in the real environment, thereby creating a nexus of the virtual and physical worlds, mixed reality systems allow for applications that occupy a physical spatial realm impossible to realise with traditional computers, or with virtual reality for that matter. Indeed, the Microsoft HoloLens, in particular, is a wearable standalone holographic computer, untethered to a desktop machine, granting unrestricted movement within the physical-digital hybrid environments it generates. The tangible hologram system put forward in this thesis uses the Microsoft HoloLens as its mixed reality component and, consequently, the discussion around Mixed Reality systems will largely focus on this technology.

One important technique employed by the HoloLens to facilitate the situating of holograms in the physical environment is *spatial mapping*, a process whereby various sensor data about the physical environment are fused to-

[1] `https://www.microsoft.com/en-us/hololens`
[2] `https://www.metavision.com`

gether, allowing the HoloLens to build up a *spatial mesh* representation of the surrounding environment. An application is thus presented with a set of spatial surfaces, important in enabling users to effectively place holograms on real world surfaces such as walls, floors, and tables[3]. Such spatial surfaces also allow holograms to be occluded by real world objects, thus adding to their perceived realism. As spatial mapping is a continuous process, the HoloLens can react to changes within a given environment or transitions between environments, while the holograms retain their position within these environments during and between sessions, doing so as long as the user desires. Consequently, the holograms appear bound to the physical world, adhering to the rules and physics of the real just as any physical object.

What can one do with such a technology? Given the capabilities just described, the possibilities are limited only to the ingenuity of those developing for the platform. Indeed, the benefits of situating interactible digital artefacts in the physical world, thereby augmenting it with digital content and new forms of interactability, are manifold. In terms of Personal Information Management, for example, one could overlay an entire virtual office on top of a physical one, augmenting physical office elements with related digital material or rich user interfaces (UIs). One might place virtual items in specific office locations, thereby leveraging the benefits of spatially arranging and organising content. In such a context, the traditional desktop metaphor need not be a metaphor and a real desktop can seamlessly become both physical and digital. Holographic application elements can be situated where they make most sense, not limited to the confines of any one space. Virtual representations of project work can be pinned to the walls around one's desk and sit alongside physical representations, while digital documents/objects may sit on the actual desktop, imbued with the capacity for those rich forms of interaction typical to digital content while still retaining the benefits of living in the real world. A virtual rubbish bin might sit at the deskside such that one may throw unwanted digital content into it. Indeed, it may be a physical bin augmented with digital capabilities, thereby allowing one to dispose of physical and virtual rubbish, as applications now exist that enable the HoloLens to recognise real world objects. The HoloLens provides for collaboration between users and so physical-digital environments can be shared when collaborators are co-located, thereby extending the benefits of working within such environments and reinforcing the perception that the holograms themselves are real entities. Collaboration can also take place over a distance, however the physical environments may not match.

[3] `https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping`

The holograms generated by mixed reality systems, in particular the HoloLens, have a compelling perceived realism, owing to those properties previously discussed. They appear, for the most part, embodied in the real world. Jansen [21], who considers embodiment as perceived, defines an embodied display as one that *"a stable and robust perception of congruence between the surfaces of the physical display medium itself and the surfaces it is meant to display, without visual or haptic discrepancies"*. Further elaborating on this definition, Jansen goes on to note that if a mid-air holographic display were out of the reach of those observing it, then such a *"'perfect' hologram would be just as embodied as a solid physical information display"*. However the illusion of such a perfect hologram would be destroyed when trying to touch it as *"the perceived congruence concerns both visual as well as tactile perception"*. These observations highlight the paradox presented by advanced mixed reality technologies such as the HoloLens. The more real something appears to be, through the properties and behaviours it exhibits, the more one wants to interact with it in *real* ways—by touching it to sense its material properties and manipulating it through direct action with the hands. Indeed, the greater the perceived realism, the greater the sense of loss when one tries to interact with it in a natural way. As promising as this current technology is, we find ourselves still removed from that which it presents—the basic human need to reach out and touch that which we directly perceive is denied to us for the moment.

The tangible hologram system seeks to circumvent this limitation by augmenting those holograms generated by *Mixed Reality* systems with haptic properties, thus allowing them to maintain their perceived sense of realism and embodiment even when touched. The motivation for doing so is self-evident and endowing holograms with tangibility has benefits beyond simply maintaining the illusion of perceived embodiment. The haptic sense is an important sensory and perceptual channel in and of itself and, as will be discussed later, can play an active role in creating effective and affective interactions with digital content, leading to highly engaging and meaningful user experiences.

While the applications of such a system, some of which will be explored further in the use case section, are many, this thesis focuses primarily on applications concerned with data exploration and analysis. The current proliferation, ubiquity and complexity of data, combined with the rate at which it is being generated, have led to a demand for more effective and innovative ways to analyse and explore it, thereby making it more accessible and comprehensible to those tasked with gaining important insights through interaction with the data. At the same time data is becoming increasingly manifest in the social fabric of society, finding representation in *data art* and *data sculptures* as well as the more traditional infographic common in many newspaper articles. Data sculptures, in particular, are of interest here as they can be seen as *physicalising* the data, bringing it into the realm of the real and conferring upon it physical properties that afford tangible interaction. Although art is, by its very nature, subjective and, in many cases, takes a form that is designed to provoke, confuse and be ambiguous, representing data in a physical way is an important idea and is of central importance to the field of *data physicalisation*.

Data physicalisation, as will be discussed in detail later, is an emerging field concerned with how data may be best represented, utilised and interacted with in physical form. Jansen et al. [23], who are important contributors to the field, have highlighted the challenges and opportunities for data physicalisation and have set out a comprehensive research agenda for the area. The tangible hologram system helps to address some of the implementation and interaction challenges associated with data physicalisation as identified by Jansen et al., thus becoming a suitable platform upon which to build effective and engaging applications concerned with exploring physical representations of data.

In some ways, the tangible hologram system is a type of Tangible User Interface (TUI) providing a haptic interface through which one can interact with and manipulate data. As such, the system complements the work done by Ishii, Ullmer and others, who are actively working within the field of TUIs. However, whereas TUIs act mainly as handles for virtual information, allowing for input through tangible interaction, tangible holograms concerns itself with both input and output, augmenting the visual output of the HoloLens with physical properties. In this vein, the tangible hologram system behaves similar to actuated dynamic shape displays such as InFORM [30] and its successor TRANSFORM [19]. However, instead of the virtual output being projected onto the device from above, it is instead seemingly projected into the real world when viewed through the lenses of the HoloLens device. Indeed, the tangible hologram system can be seen to work in tandem with technologies such as TRANSFORM, seeking to approximate the *programmable matter* of Ishii's Radical Atoms vision [18].

1.1 Problem Statement

The recent emergence of affordable and accessible digital fabrication technologies, such as multi-material 3D printers, have made the process of prototyping and producing relatively complex static data physicalisations much more straightforward [23]. However, the range of physical properties that may be conferred on such physicalisations is limited by a lack of suitable materials and ways of accommodating such materials in the fabrication process [23]. Additionally, in terms of dynamic physicalisations, a sizeable proportion of the technologies currently applicable to the physicalisation of data require significant supporting infrastructure, thus rendering such approaches largely immobile. For certain applications, the resultant restricted working area may constitute a substantial disadvantage from a usability point of view. This issue, coupled with the fact that such technologies are expensive to prototype and, in many cases, difficult to implement, makes the realisation of dynamic data physicalisation a challenging prospect for those working in the area.

Jansen et al. [23] also note several interaction challenges that need to be met if data physicalisation is to be successful in providing for the effective and engaging exploration and analysis of data. Among these challenges is the necessity for the prompt and smooth reconfigurability of physicalisations and the need to combine physical and synthetic interactions in an appropriate and effective manner. Those working within the area of Tangible User Interfaces (TUIs) have pioneered the bridging of the physical and virtual in novel and innovative ways, designing applications that afford highly engaging and compelling interactions in the process. The Radical Atoms vision now seeks the seamless integration of the digital and physical, accomplished through a future dynamic programmable matter. This vision has already inspired a number of remarkable new technologies, however, the material itself, and the rich interactions that it might grant, remain largely out of reach. Indeed, data physicalisation would benefit greatly from the the attainment of such a material as it would help in overcoming many of its implementation and interaction challenges. Given the problem statement just presented, then, the objectives of this thesis can be stated as follows:

- **To investigate** whether TUIs, and interactions therewith, can be effectively employed to augment and enhance data exploration and analysis tasks.
- **To explore** how to best simulate the programmable matter described in the Radical Atoms vision, as set out by Ishii et al., using existing technologies.
- **To develop** a platform to help address the implementation and interaction challenges facing those working within the field of data physicalisation.

1.2 Contributions

The work undertaken in this thesis has grown from an initial exploration into tangible user interfaces, and how they might be employed in data analysis and exploration tasks, to how one might approximate the vision of Ishii's programmable matter using existing technologies—thereby allowing those working in the area of data physicalisation to explore the richness of possibilities such a material may have to offer. To evaluate the viability of such an endeavour, we developed the Tangible Hologram (TangHo) system, and, in so doing, learned much that may be beneficial to those wishing to further the work. With this in mind, then, the main contributions of this thesis are:

- The **design and development of a novel mobile tangible hologram system**. As mentioned, 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.
- The **production of a digital model of the prototype construction**, which was largely developed with the Lego Mindstorms system, such that others may easily construct their own version—one they can further develop and refine, thereby building on the work already undertaken.
- An **exploration of those areas** most relevant to the development of a data physicalisation simulation platform. The discussion on the main themes related to these areas can be found in the background section.
- A number of **use cases** to illustrate potential interactions with the tangible hologram system. These use cases may serve as a useful starting point for those new to the concept of physically augmenting holographic content.

1.3 Methodology

The Design Science Reseach Methodology, as described by Peffers et al. [39] was used to inform the nature and order of the work undertaken in this thesis.

The approach consists of a number of activities designed to effectively guide one through the research process. The initial activity involves identifying a given research problem and *"justif[ying] the value of a solution"* [39]. The problem statement, as presented in the problem statement section above, identifies those issues we wish address and the many benefits in addressing these issues is clearly motivated. The objectives of the thesis are then derived from the problem statement.

Another activity of the methodology involves the design and development of an artefact. As noted by Peffers et al. [39], this activity includes *"determining the artifact's desired functionality and its architecture and then creating the actual artifact"*. This process has indeed been followed here. From an initial investigation into what requirements a given solution would need to satisfy, we ideated, designed and iteratively developed the tangible hologram system prototype. A demonstration of the TangHo system's applicability in addressing the issues raised in the problem statement is provided through a number of use cases, which are presented in chapter 6. Finally, we have already begun the process of communicating the concept of a tangible hologram system, the motivation behind its inception and the novelty of its approach in addressing some of the implementation and interaction challenges associated with data physicalisation, by publishing an early paper on the work.

1.4 Thesis Outline

Chapter two gives the background necessary to contextualise and motivate the work undertaken in this thesis. The field of data physicalisation and the benefits of physicalising data are initially explored, as these provide the main motivation for developing a system capable of simulating such physicalisation. Haptics, a chief concern for the tangible hologram system, is then covered, with the discussion centring on human haptic perception and haptic interfaces. Next, the focus shifts towards the work carried out by Hiroshi Ishii and others on the subject of Tangible User Interfaces (TUIs) and the Radical Atoms vision. These areas merit exploration as they share some concerns with data physicalisation and can be seen as providing possible current and future enabling technologies for the field. Indeed, the notion of programmable matter, as set out in the Radical Atoms vision, was a key inspiration for developing the tangible hologram system and the platform can be seen, in many ways, as an attempt to approximate programmable matter using existing technologies. Finally, an analysis of the Microsoft HoloLens, the mixed reality component of the tangible hologram system, is carried out and this should serve to justify its selection over other options.

In chapter three, work related to the tangible hologram system is described and a comprehensive description of the tangible hologram solution is presented in chapter four. Some of the key issues surrounding the design of the platform, and how these issues were dealt with, are discussed here. In addition chapter four details how the system helps to address some of the implementation and interaction challenges currently facing those wishing to work within the field of data physicalisation. Chapter five, then, provides an account of how the solution was implemented and this includes implementation details for both hardware and software components. A number of use cases illustrating the envisaged use and potential benefits of the tangible holograms system are set out in chapter six. Here a mixture of data-centred and alternative applications of the platform are presented. Finally, potential future work for that tangible hologram system is explored and a summation of the work undertaken is made.

2
Background

This background section aims to provide some context to the tangible hologram system, helping to situate it relative to other fields of endeavour. The conception and development of the platform were borne out of research into the areas discussed here and an exploration of their central themes, goals and challenges should provide some motivation regarding the need for and benefits of developing a solution capable of data physicalisation simulation.

2.1 Data Physicalisation

Data physicalisation, techniques and methods for achieving physicalisation and interactions with and between physicalisations constitute an important concern for this thesis as the tangible hologram system seeks to overcome many of the technical limitations associated with the area. The field of data physicalisation is a relatively recent development but it has already garnered a fair degree of interest and a number of studies into investigating the effectiveness of physicalising data [put the refs here] have been undertaken, resulting in some promising outcomes. In setting out the challenges and opportunities for data physicalisation, Jansen et al. [23] define a data physicalisation as *"a physical artefact whose geometry or material properties encode data"*. As such data physicalisation is closely related to the more mature field of Information Visualisation and, indeed, there is a certain amount of overlap between the two, particularly in terms of visuospatial encodings. One way in which data physicalisation can offer some advantages over traditional visualisation is in its ability to extend the range of encodings to the haptic sense. Temperature, weight, texture, density, inertia and so on, may be used to add extra dimensions to representations that were previously restricted to a palette of visual encodings. However, it must be noted that the effective mapping of data features to physical variables is no small task and this issue is of major importance to the field. In discussing the issue, Jansen et al. [23] state that *"Identifying, exploring and classifying physical variables is a research challenge that will be the key to understanding the design space of data visualizations"*. Work on exploring these mappings has already begun with one study [24] investigating the effectiveness of size as a physical variable.

Noting that data first manifested itself in a physical form before the advent of paper and screen, the abacus being a case in point, Jansen et al. propose that, given recent advances in 3D fabrication and tangible computing technologies such as actuated shape displays, the opportunity now exists to bring data back to the physical realm in purposeful and innovative ways. Indeed, the notion of data physicalisation as presented by Jansen et al. is timely given the current proliferation of data, the ever increasing rate at which it is being generated and its growing importance and utility in areas outside of those traditionally data-intensive fields. The ubiquity of data can be evidenced in the many ways it is weaving itself into our everyday lives, from smart data-generating devices to aesthetically pleasing infographics that seem to have acquired a state of omnipresence. Data is also finding representation and expression outside of traditional forms, venturing into areas such as fashion and the arts [24, 49]. All of this serves to create an appetite for new, engaging and effective ways to explore and analyse data and data physicalisation offers this promise.

2.1.1 The Benefits of Data Physicalisation

The main goal of data physicalisation is to provide for richer, more engaging and effective interactions with data through the utilisation of *"computer-supported physical data representations"* and Jansen et al. provide an outline of the benefits of such physicalisation. These include the exploitation of our natural perceptual exploration skills, including active perception, depth perception and the use of non-visual senses, the intermodality of physicalisations and their accessibility and the cognitive and affective benefits of intacting with physical artefacts [24]. The areas of haptic perception, along with the cognitive and affective implications of interacting with physical artefacts, will be discussed in further detail later. However, at this point, it would seem that there are several advantages to the endeavour.

Despite the apparent benefits, some authors, such as Wiberg [47], warn about that there are risks associated with the move away from traditional representations to physical and material forms. In *"over-privileging the authenticity of the non-computational world"* and placing too much emphasis on the implementation of the move to the material simply on the grounds that this makes information more *"real"* comes at the expense of considering questions about interaction with such manifestations, which should be of primary importance [47]. Indeed, this is a valid point and a number of studies into the effectiveness of physicalising data [44, 43] have noted shortcomings in how interactions with physicalsations were designed, in some instances describing them as being *"enforced"*, non-productive and lacking in engagment. Wiberg also notes that the march to physicalisation might see the *"abandoning of some of the intellectual gains made through earlier critiques of representation"* [47].

There are, no doubt, several risks in moving towards the physicalisation of information without taking into due consideration matters concerning interaction and effective ways of re-representing data in physical form. It is important to reap the benefits provided by physicalising data without losing those lessons learned from previous representations in the translation. That being said, Jansen et al.'s presentation of data physicalisation is measured and their proposed research agenda considers both implementation and interaction, in addition to understanding what is effective and how physicalisations may be suitably evaluated [23]. As previously mentioned, a number of evaluations into the effectiveness of physicalisations have already been undertaken.

One such study [22] found that, in terms of information retrieval, physical 3D bar charts outperformed on-screen represention of 3D bar charts—one rendered in mono and one stereoscopically with the aid of 3D glasses. The researchers note that this evidenced efficiency may be owing to the *"visual realism"* of the physical bar charts and that the ability to touch the physicalisations, thereby allowing participants to *"mark"* parts of the physicalisations with their fingers, could be seen as providing participants with the capacity to use *"external cognitive and visual aids"* [22]. Researchers in another study assessing the effectiveness of 3D physical bar charts in information recall [43], this time when compared to 2D representations, also noted that the extra dimension present in the 3D physicalisations afforded them additional *"distinctiveness"* and this may have played a role in their outperforming the 2D representations in terms of memorability.

2.2 On Haptics

The tangible hologram system provides a haptic interface whereby users can touch holograms, allowing them to be explored haptically as well as visually, and, therefore, the area of haptics plays an important role in this thesis. The term *haptic* can have a physiological or technological meaning depending on whether a haptic stimulus is being sensed by a human or rendered, or, for that matter, sensed, by a machine or device. The Oxford English Dictionary[1] defines haptics as *"relating to the sense of touch, in particular relating to the perception and manipulation of objects using the senses of touch and proprioception"*. While this definition would appear to relate exclusively to the human sense of touch, several of the example sentences associated with the definition contain terms such as *haptic force-feedback*, *haptic sensors* and *haptic devices*, suggesting strong technological associations as well. Indeed, the area of haptics, in terms of human-computer interaction, concerns itself with both of these aspects and when designing haptic systems that allow for effective and rich forms of interaction, one has to carefully consider both the physiological and technological side of haptics and how these relate to each other.

2.2.1 The Human Sense of Touch and Active Perception

The human sense of touch, comprising two subsystems, namely the cutaneous, or tactile, modality and the kinesthetic, or proprioceptive, modality, is highly effective at sensing and processing the properties of physical objects [28]. While the tactile modality concerns itself with detecting low-level forces, important in sensing material properties of objects, such as texture and temperature, kinesthesia and proprioception deal with larger forces detected by mechanoreceptors in the muscles and joints, with proprioception providing a sense of position of the limbs and body in space and kinesthesia providing a sense of movement of those limbs. These larger forces are important in detecting properties such as weight and inertia. In interacting with the physical world the information from both of these subsystems is integrated, in numerous ways, to provide haptic perception [28].

Gibson [13] notes the importance of touch as a perceptual sense, not merely a passive or receptive one. In terms of haptic perception, then, a distinction can be drawn between active and passive perception with the former being an exploratory activity, undertaken with purpose, where one generates

[1] `https://en.oxforddictionaries.com/definition/haptic`

the stimulation through interaction with the environment, and the latter seeing external forces, be they physical objects or environmental conditions, acting on the sensing party [13]. The effectiveness of touch as a perceptual sense is evidenced in peoples' ability to recognise familar objects quickly and accurately through touch alone [28]. Gibson [13] also demonstrated the efficacy of active touch with his *Cookie Cutter* experiment where people were asked to distinguish between a number of differently shaped cookie cutters without looking at them. When actively exploring the objects particpants correctly discriminated between the different shapes 95% of the time whereas when the cookie cutters were pressed into the palm of the hand, so that participants were sensing the object passively, that dropped to 29%.

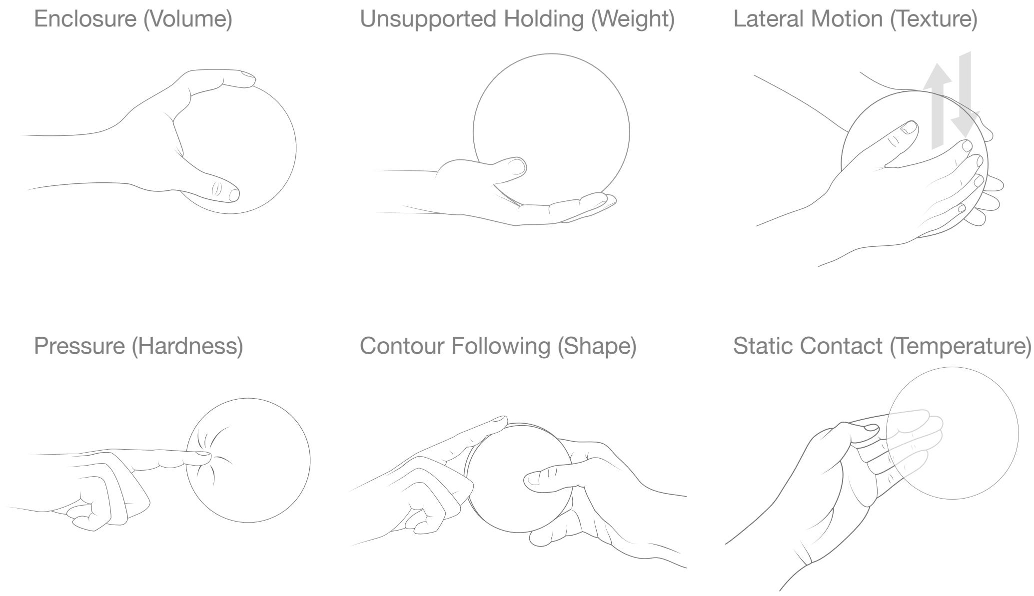


Figure 2.1: Six manual exploratory procedures adapted from [27]

Through the active perception of objects and the physical environment, then, touch becomes an important sensory channel through which to recieve and process information and such a channel is of great importance to the field of data physicalisation. Lederman and Klatzky [27] identify six exploratory procedures, which they term a *"stereotyped pattern of manual exploration"* commonly employed in haptic perception, either with or without vision, and these are illustrated in Figure 2.1. Such exploratory procedures will be greatly utilised when exploring data-driven environments and data physicalisations and the tangible hologram system, when further developed, should be able to provide for each of these procedures.

2.2.2 Haptic Interfaces

Haptic interfaces are, in essence, devices that allow for Human-Computer Interaction (HCI) through touch, thus they can leverage the benefits of haptic perception just discussed. Many useful descriptions of haptic interfaces exist that can serve to eludicate some of the central ideas behind haptic systems. For example, Salisbury et al. [40] consider haptic devices as *"small robots that exchange mechanical energy with a user"* and use the term *device-body interface* to *"highlight the physical connection between operator and device through which the energy is exchanged"*. Biggs and Srinivasan define the general term *'haptics'* as *"umbrella term covering all aspects of manual exploration and manipulation by humans and machines, as well as interactions between the two, performed in real, virtual or teleoperated environments"*. They go on to define haptic interfaces as those that *"allow users to touch, feel, and manipulate objects simulated by virtual environments (VEs) and teleoperator systems"*. Another useful definition comes from Hayward et al. [14], who define haptic interfaces as *"being concerned with the association of gesture to touch and kinesthesia to provide for communication between . . . humans and machines"*. At this point, an important distinction can be made between active and passive haptic interfaces. As the terms suggest, active interfaces *"present controlled forces to the user, allowing him or her to feel virtual objects as well as control them"*, while passive interfaces, such as the traditional mouse and keyboard, in contrast, simply detect the user's hand movements [3].

With these definitions in mind then, haptic interfaces can be seen as allowing, in many cases, for the bidirectional exchange of energy, or information, between a human agent and a computer, although in some cases interfaces may be unidirectional. Focussing on the bidirectional case, then, changes to an underlying computational model can be reflected in the haptic feedback rendered to the user, while the user may also effect changes in the model through manipulation of the same interface. Figure 2.2 illustrates this exchange of energy information. As can be seen from the figure, there are two main components to this information loop, the human agent and the machine. There is an evident parallel relationship between the elements of both cycles within the loop, with the human and machine components both having sensors, actuators and a processing unit. Forces from one are sensed and processed by the other and any requried actuation response commands are calculated and passed to actuators to execute them. Salisbury et al. [40] describe a typcial haptic rendering loop as following a sequence where the physical position of haptic interface's joints are first sampled and passed to control algorithms which use the information to calculate the the position of the device-body interface in Cartesian space. The device-body interface in

Cartesian space is a virtual represntation of the haptic interface, a so-called *"avatar"* [40]. A collision detection algorithm determines if the avatar has collided with any virtual objects and returns the *"degree of penetration or indentation"*. Finally force-response algorithms compute interaction forces based on this information and the haptic devices is actutated to render these forces onto the user's hand [40].

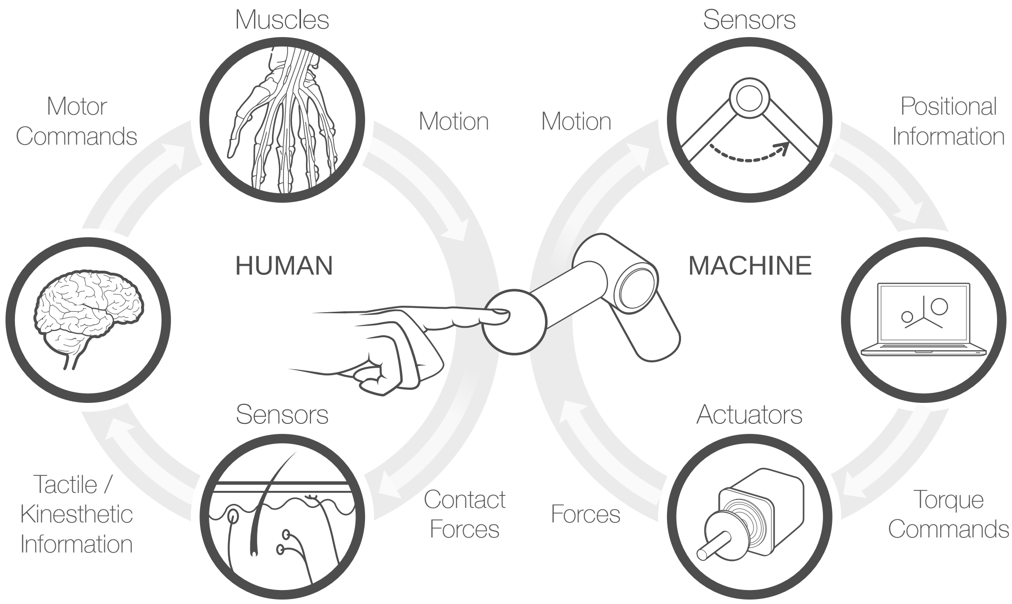


Figure 2.2: Bidirectional haptic rendering system overview

While the illustration in Figure 2.2 depicts user interaction with an actuated articulated robotic arm designed to provide haptic rendering, the form haptic interfaces take can vary greatly depending on the requirements of their target applications. These forms naturally lend themselves to supporting a varying range of exploratory procedures, which, as previously mentioned, are important in active perception. Haptic interfaces can be differentiated from each other in many different ways and, as noted by Salisbury et al. [40], when classifying haptic interfaces, one might consider their grounding locations, their intrinsic mechanical behaviour or the number of degrees of freedom (DOF) of *"motion or force present at the device-body interface"*. Another way of grouping of current haptic technologies might be to consider ground-based devices, body-based devices and tactile displays as distinguishable clusters of haptic interfaces [10]. In terms of ground-based devices, haptic interfaces such as manipulandums or actuated articulated robotic arms, are grounded to a fixed point such as the surface of a desktop or table, thereby allowing them to haptically render physical properties such as weight and inertia, at the expense of mobility. Other haptic devices, such as force-feedback gloves and exoskeletons, may be worn and so these devices can be considered grounded with respect to the user—if the exoskeleton itself is not grounded to some fixed point. Thus these body-grounded or body-based haptic devices grant mobility. The range of exploratory procedures supported by these devices, however, depends on their type. A force-feedback glove will support enclosure to enable the detection of a virtual object's global form and volume and pressure may be applied to determine its relative stiffness, but the device can not determine an object's weight through unsupported holding or haptically render an inert object. It is possible for body-grounded exoskeletons, on the other hand, to convey the weight or inertia of a virtual object, but they may not support those exploratory procedures permitted by force-feedback gloves. One of the most common forms of haptic device is the relatively inexpensive hand-held game controller used with most game consoles and VR headsets. Here vibrotactile feedback, typically generated by eccentric rotating mass (ERM) vibration motors, is used to signify contact/collisions with virtual content or convey urgent game information, such as rapidly diminishing health, in a non-visual way. However, the range of exploratory procedures supported by devices such as these is relatively limited. A more detailed discussion of some of these haptic interfaces will be undertaken in the related work section.

Haptic devices have a wide range of applicability outside of their more obvious deployment in gaming and entertainment. Indeed, haptic interfaces are used in areas such as scientific discovery and engineering where haptics can help to *"convey the existence of small details, which typically clutter the graphical presentation of data"*, in the case of the former, and allow designers to *"experience minute details with their hands, such as wanted or unwanted artefacts of a design which are cumbersome to display visually"*, in the case of the latter [14]. Haptic interfaces are also commonly used in medicine for training surgeons via surgical simulators, thereby reducing the costs associated with such training, and in telesurgery, to aid in the manipulation of robots performing minimally invasive surgery [3]. Indeed, the topic of using haptic feedback to enhance force skill learning was explored in work by Morris et al. [33], which suggested that *"in conjunction with visual feedback, haptic training may be an effective tool for teaching sensorimotor skills that have a forcesensitive component to them, such as surgery"*. Haptic devices can also play a beneficial role in patient rehabilitation, acting as an important component in therapies designed for those recovering from a stroke, for example [11, 5].

It is evident that haptic interfaces represent a versatile, diverse and powerful group of devices which can be effective perceptual tools in their own right, or can be successfully employed with other modalities to enhance the user experience. They are regularly employed in applications designed to address the needs of a wide variety of users, be they concerned with entertainment, science, engineering, professional training or medicine. The tangible hologram platform seeks to add the area of data physicalisation to the aforementioned list and, in so doing, allow for the exploration, testing and evaluation of physicalised data representations, and interactions therewith, through the haptically augmented simulation of such physicalisations.

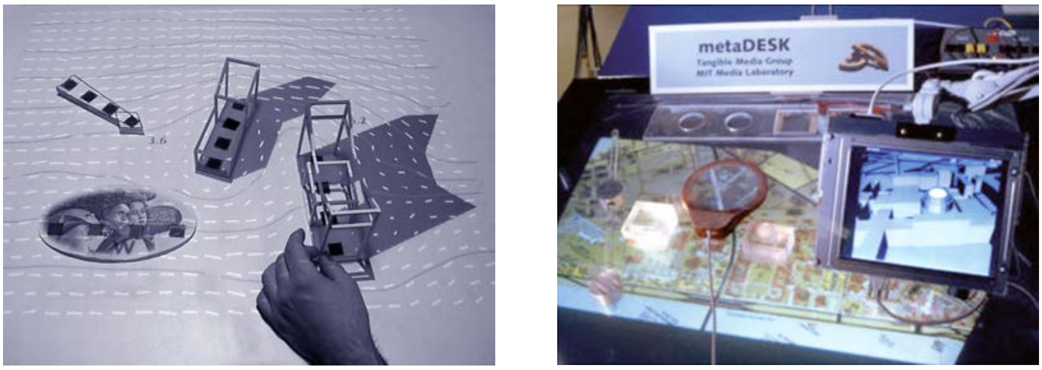
2.3 Tangibility and Tangible User Interfaces

The area of Tangiblie User Interfaces (TUIs) is an important one to consider for those wishing to develop a data physicalisation simulation platform as the area can be seen to have some overlapping concerns with that of data physicalisation. In light of this, a brief overview of the area will be set out here. Hiroshi Ishii and Brygg Ulmer, both promiment contributors to the area of TUIs, set out an often cited vision for TUIs in Tangible Bits [20] and the discussion around TUIs will largely center on this vision. TUIs can be seen to act as *"physical manifestations of computation"* [18], thus they amount to physical handles through which one can interact with and manipulate virtual content. In contrast to traditional input modalites, such as mice and keyboards, which may be seen as more generalised artefacts mediating interaction with GUIs, the tangible interfaces advocated by Ishii and others can be thought of as having a more direct relationship with the underlying data they are coupled to. One of the key concepts in the Tangible Bits paper is the *"coupling of bits and atoms"*, which involves the *"seamless coupling of everyday graspable objects (e.g. cards, books, models) with the digital information that pertains to them"*. If this seems to resonate somewhat with Mark Weisner's vision of *Ubiquitous Computing*, it is because the authors were inspired by the idea. However, they make a clear distinction in noting that their aim is to *"[awaken] richly-afforded physical objects, instruments and spaces to computational mediation"* [20]. It is evident then that the physical form of the tangible artefact, and its provision of those affordances specific to accessing, comprehending and manipulating that digital information to which it is bound, is of key importance. Ullmer and Ishii, in following up on their inital work in Tangible Bits, cite the abacus as a *"compelling prototypical example"* of a TUI [45]. They note that the abacus does not differentiate between what is input and what is output, rather the arrangement of beads and rods constitute *"manipulable physical representations of numerical values and operations"*, while in tandem acting as *"physical controls for directly manipulating their underlying associations"*. When one considers the underlying principle behind this analogy, one can begin to draw some clear parallels between the concerns and motivations behind TUIs and a number of important concerns in the area of data physicalisation.

Indeed, the field of data physicalisation may learn a lot from the development of TUIs. In their efforts to bridge the physical and digital, and remove the distinction between representation and control, those working on tangible interfaces have created artefacts with rich affordances, while also designing meaningful interactions with and between them. According to Ishii et al. [18], *"tangible design expands the affordances of physical objects so that they can support direct engagement with the digital world"*, thereby leveraging on our natural perceptual abilities to *"make information directly manipulable and intuitively perceived"*. This, again, would appear to align with some of the goals of data physicalisation. At this point, it is worth looking at some notable TUIs to illustrate how such interfaces provide for novel, yet purposeful, interactions with digital information.

An early and well-known tangible interface is an urban planning system called URP, which creates a space designed to simulate environmental factors such as windflow and how shadows are cast at different times of the day [45]. These factors are important when trying to optimally position and orient buildings so as to reduce the effects of wind, glare and shadowing on neighbouring buildings. The interface takes the form of a workbench, with the central tangible components consisting of scale architectural models of buildings and a number of physical tools to aid in controlling those simulated aspects of the system. As a user repositions and reorients a building on the workbench surface through direct tangible manipulation, a projector, positioned overhead, updates the shadows cast by that building at the specified time of the day. The time of day can be changed by adjusting the hands of the *"clock tool"*. In addition the resultant change in windflow is shown if the user places the *"wind tool"* on the workbench surface and this tool can be oriented to see the effects of wind from different directions. The material of buildings can also be changed with a *"material wand"* with the effects of the change being reflected in the simulation [45].

As one can imagine, this type of tangible interaction creates a very immediate and involved experience for those using it. In terms of navigating the space, one can simply walk around the workbench to get different perspectives and move closer to the architectural models to get a close up view. The URP system facilitates collaborative exploration and analysis, as multiple users can update various parts of the model in synchrony and collaborators share the same generous view of the simulation. The tools employed to control the simulation take physical form, embodied versions of tools that have long since been relegated to the of confines of the icon in traditional GUI applications. Just as tools have a specified function in the real world, each physical tool in the URP system has its own purpose. The benefits of this are that the tools can be seen to hold the state of that information to which they are bound, thereby acting both as a means of representing the information and controlling it—an important capability for Ishii and Ullmer. Consider, for example, the aforementioned clock and wind tools. The clock hands stay at their specified configuration and the wind tool remains in its oriented position until a change is effected by the user. In both cases, cues about underlying state make such knowledge explicit and immediate. This is in contrast to more generic input devices such as the mouse, which can be seen as offering the temporally multiplexed manipulation of digital content, thus rendering them incapable of representing the state of any one entity.



(a) URP © MIT (b) MetaDESK © MIT

Figure 2.3: Tangible User Interfaces developed at MIT Media Labs

Being physical, these tools can also offer rich affordances for interaction, thereby leveraging all the benefits of physicality, while still being coupled to digital content. Indeed, the interplay between tools is also quite and important and a good illustrative example of this is how one might use the *"phicons"* of the *"metaDESK"* research platform [20]. With the metaDESK, one can use a wide range of tangible elements to explore digital information, including physical icons or phicons. In demonstrating the utility of the platform, Ishii and Ullmer display a map of the MIT campus on the desk surface, allowing one to place a phicon, representing a campus landmark, on the metaDESK. Doing so positions the map view so as to align it with that phicon. Should a second landmark be placed on the desk surface, thereby giving the system two points of reference, the view changes so that the map is now aligned with both phicons. Pushing or pulling the phicons closer to or farther away from each other now acts as a zooming operation, while rotating one phicon relative to the other allows for a rotation of the whole map. Indeed, these operations can be performed in tandem. The interplay between these tools feels natural and intuitive, resulting in effective interactions with the underlying digital content.

To gain some insight into how tangibles may be used in a data exploration and analysis context, one can look at the Tangible Query Interfaces implemented by Ullmer et al. [46]. One adopted approach uses *"physically constrained tokens to express, manipulate, and visualize parameterized database queries"*, thereby *"extend[ing] tangible interfaces to enable interaction with large aggregates of information"* [46]. In essence, queries on the underlying data sets can be constructed by placing the tokens, which can be bound to database parameters, into so-called *"query-racks"*. In this way, queries with multiple parameters may be built up. Additionally, the physical manipulation of these tokens *"modifies parameter thresholds, expresses Boolean relationships, and controls visualisations of query results"*. A nice feature of the tokens and constraints approach is that the physical constraints imposed on the placement of tokens affords a communication of *"the kinds of interactions the interface can (and cannot) support"* [46]. This has implications for the design of data physicalisations, as, in many cases, a data physicalisation will have to rely on the affordances granted by its physical form to express and communicate the available and appropriate actions for operating on it.

TUIs may also be effectively employed to foster learning and understanding in a classroom setting [38], and Jansen et al. suggest this approach may help students engage with and comprehend data and data representations in a non-visual way [23]. Others suppose that the success of such approaches may suggest that *"a physical representation of information could generate a more complete or detailed spatial representation in [a] subject's memory"* [43]. In their literature review in learning with tangible technologies, O'Malley and Fraser [38] note that benefits for learning from tangible interfaces have been suggested by psychological and educational research and that well-conceived activities with tangibles can help children of all ages overcome the difficultes associated with more traditional and abstract representations when performing symbol manipulation tasks. They highlight the fact that the physical activity itself plays an important role in building the *"representational mappings"* important in providing a grounding for further understanding later on [38].

It is evident, then, that the area of data physicalisation can draw much from the development of tangible interfaces, both in terms of how physicalisations may be manifested so that they afford rich user interactions, and how the physicalisations themselves might effectively and coherently interact with each other. Through interchangeable and special-purpose end-effectors, the tangible hologram system, too, may emulate some of the benefits of tangible interfaces. Indeed, if the end-effectors were capable of affording many complex interactions, the physical tools of tangible interfaces could be simulated while still providing the benefits of combined representation and control as mentioned above. Given that a user can only hold one or two tools in their hands at once, this is entirely plausible.

2.4 Programmable Matter

Ishii's [18] *Radical Atoms* paper further extends the idea of tangible user interfaces, shifting the paradigm from interacting with virtual artefacts through physical handles and controls to physicalising such virtual artefacts completely in the real world. Digital objects, once decoupled from the physical handles that controlled them, now become directly manipulable, acting as both input and output and offering up exciting challenges for the field of interaction design and, importantly, data physicalisation. The central idea of Radical Atoms, which allows for *"vision-driven"* research into these new forms of *"human-material-interaction"*, is a hypothetical and futuristic programmable material, a so-called *"digital clay"* capable of dynamic reconfiguration and direct user interaction. The vision of Radical Atoms overcomes some of the limitations of tangible user interfaces, namely their decoupled relationship with the virtual objects to which they are bound and, as Ishii notes, their limited capacity to change their form and properties to reflect changes to virtual representations. Dynamic affordances to allow for appropriate and purposeful interactions are also somewhat limited.

One can draw an analogy with a Model-View-Controller (MVC) when describing Ishii's Radical Atoms concept and the envisaged interaction with Radical atoms is illustrated in Figure 2.4. In Ishii's terms the material is bidirectionally coupled to an underlying digital model such that changes in the physicalisation result in changes in the digital model and visa-versa. Some important requirements for the material are listed to ensure that such bidirectionality is achievable. In the first instance the material should transform its shape according to the requirements for user interaction and to changes in the digital model it represents. In addition, the material should inform the user of its affordances and, given that the interface may be in a constant

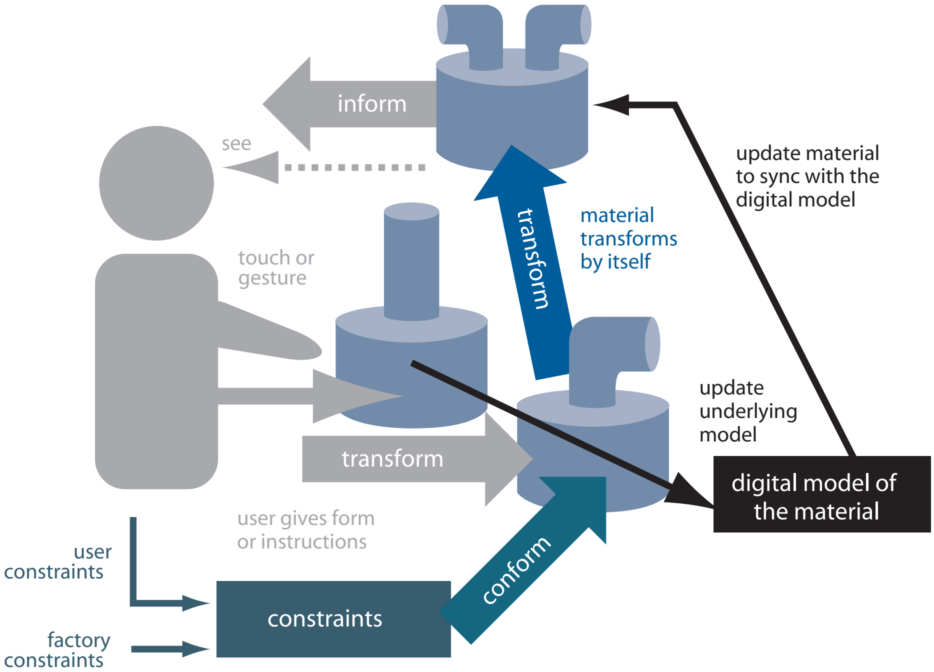


Figure 2.4: Interactions with Radical Atoms © Ishii et al.

state of flux, such affordances would need to be dynamically rendered. Indeed, Follmer [12] has already begun to explore the area of dynamic physical affordances using the InFORM 2.5D shape display, achieving some interesting results in the process.

Given the potential of Ishii's programmable matter, it is evident that its realisation would be of great benefit to the field of data physicalisation. Jansen et al. [23] note that programmable matter, along with recent technologies, such as shape displays, have the capacity to radically change *"the fidelity and flexibility with which data can be made physical while decreasing the cost to do so"*, adding that such materials and technologies constitute an *"unprecedented opportunity to create new forms of physical representations"* that will eventually be capable of rivalling traditional desktop computers in terms of complex data analysis. Indeed, in illustrating the potential of programmable matter to help realise the goals of data physicalisation Jansen et al. [23] describe a convincing usage scenario for neurosurgical planning where a surgeon interacts with an artefact composed entirely of programmable matter. It is worth noting that while the idea of programmable matter may seem a distant reality those current technologies mentioned by Jansen and Ishii, such as shape displays, micro-robotics and anti-gravity interaction elements, can be viewed as constituting the first steps towards realising this vision.

2.4.1 Data as Material

In thinking about data in terms of programmable matter and how it relates to data physicalisation, it can be useful to consider data a material in and of itself, in a manner similar to how many within the HCI community seek to describe the computer as material in the wake of the development of new computational materials [47]. Data has characteristics and features that can be mapped to different physical variables and so, when physically rendered, the data, or at least certain aspects of the data, can be seen to constitute a unique material in the real world. It can be acted upon and take on new forms. In effect, this data material is the stuff that the physicalisation is made from and can communicate as much about itself as the form the actual physicalisation takes. It is this data material which the programmable matter or *"physical-digital substrate"* must emulate faithfully and with high fidelity.

This data material can be acted upon like any other material, within what the properties of the material will allow and these properties are conferred upon it by the data itself, for example a sparse data set might confer the property of sponginess. As the data material is manipulated, its properties and affordances may be altered as a result. For example, a fluid material might firm up and become more rigid, connoting that the representation is less open to manipulation at this point. Another example may be a hot material cooling down as a result of certain interactions indicating that, in this altered state, the data representation has become more stable and lost some of its dynamism (this could also be denoted through a colour change). However, as previously mentioned the effective mapping of data features to physical variables is challenging and needs careful consideration.

Wiberg [47], in discussing computers as a material, notes that when we treat computers as *"just another material"*, then *"instead of asking how should it work and look, we might now find ourselves asking, what kind of material is it?"*. The same can be said of treating data itself as a material and viewing data as a raw material might be a useful tool for thinking about how it might be manipulated and interacted with. Here analogy may play a useful role. Materials with different properties lend themselves to different types of manipulations and are worked upon by practitioners in different ways. Clay lends itself to additive/subtractive processes and, depending on how much it has been exposed to air, is malleable and plyable. When exposed to air for an extended period, it loses a lot of its plasticity but forms can then be refined through subtractive methods. Stone and wood on the other hand are manipulated through subtractive methods, although wood can be joined together by other means (e.g. glue and nails).

Working with data as a material in the ways described above is currently not possible as it requires programmable matter capable of rendering physical properties such as weight, texture, temperature, density and so on. In time, and with further development, it is envisaged that our tangible hologram system will be able to generate these physical properties through a combination of refined force feedback rendering and sophisticated end-effectors. Indeed, Araujo et al. [1] have incorporated many of these capabilities into their *Snake Charmer* solution and this work will be discussed later in the related work section.

2.5 Affective Interactions with Data

In a study comparing the effectiveness of 2D and 3D physical data representations in terms of information recall, Stusak et al. [43] note that the benefits associated with physical visualisations, namely the tangible nature of physicalisations and the leveraging of natural spatial perception abilities when interacting with them, are heightened when the data itself is more *"comprehensible"* and meaningful. This insight was derived from the fact that the overall recall performance of both the 2D and 3D physical representations was significantly lower when the data, in this case economic data, was uninteresting or less meaningful, with no real performance difference between the two modalities being evidenced. In contrast, data that was perceived as more relatable, in this case population data, saw an increase in the performance of both the 2D and 3D physical modalities with the latter outperforming the former in terms of data recall accuracy.

The important point to note is the significant role that the underlying data set played in terms of memorability and information recall. In Stusak et al.'s study, which was designed to make a direct comparison in performance between paper-based 2D bar charts and wooden block-based 3D bar charts, the data took on a traditional form, namely the bar chart. This constituted, in effect, a simple translation of a well-known 2D data representation into a 3D one, and doing so carries a lot of merit. This approach, which can be seen in many similar investigations into data physicalisation, allows a study to focus on the effects that the physicalisation of data, or certain characteristics thereof, has on certain performance metrics such as comprehension and, in this case, recall of information, by making comparisons with 2D counterparts. Keeping the form of the data simple and familiar helps to facilitate the goal of the study. In addition Jansen et al. [23] note that if physicalisations are to be *"readable"*, in contrast to the sometimes subjective nature of data sculptures, then *"a principled way of encoding data"* needs to be formulated. This is an important point and tried and tested forms of data representation, which should not be overlooked or discarded, seem like a good place to start. However, as also noted by Jansen et al., traditional 2D representations are purely visual and designed to be looked at, not touched or interacted with in a physical way, and while some already established visual encodings may be entirely suitable for physicalisations, direct translations may not always be enough.

The downside of the aforementioned approach employed by Stusak and others, is that interactions with the physicalisations can, in certain circumstances, be contrived and non-engaging and in many ways this is a big challenge for the field of data physicalisation. The affordances that the physicalisations provide for meaningful and engaging interactions, in addition to providing for interactions that allow thorough comprehension and accurate information retrieval from data sets, is important. Indeed, Stusak et al. noted the difficulty experienced in designing effective interactions with the physicalisations, noting that the *"enforced"* interactions of assembling/disassembling and grasping the physical bar charts was *"not productive"*. In a separate study on the memorability of physical visualisations [44], this time comparing 3D physical bar charts to 2D representations on an Apple iPad, Stusak et al. again note the interaction challenges associated with these physicalisations, stating that participants *"did hardly explore it haptically"* and that *"the use of a more handy physical visualization or tasks that more clearly require haptic interaction could change this in further studies"*. That being said, the study again found that the 3D physicalisations outperformed their 2D counterparts in terms of memorability. The researchers go on to note that physicalisations that invite more interaction should *"increase the observed effect"*, thereby making these physical visualisations even more memorable.

It is evident that the memorability of data is bound to how effectively the manifestation of such data, be it physical, auditory, visual or otherwise, affords meaningful interactions. The more engaging and relatable interactions with data are, the more powerful and memorable the experiences generated become, thus rendering the data itself more memorable. Data that is not brought to life in a meaningful way will lead to effects such as those reported by Stusak et al. [44, 43], whereby the underlying data sets play a much larger role in affecting performance metrics, with the benefits of physicalisations themselves varying between such data sets.

Designing engaging experiences with data through physicalisation can have a positive impact on other concerns outside of memorability and information recall. Studies concerning what Wiberg [47] terms *"computational expressivity"*, where digital-physical interfaces consider *"expressive dimen-*

sions" and designers *"advocate the evocative and pleasurable as core to the experience of computation"*, can yield some very important insights for data physicalisation and human-material interaction. In their study on peoples' affective responses to data, Hogan and Hornecker [16] note that representation through different modalities can elicit different affective responses to the same data set, thereby generating different user experiences. During the study participants were recorded as they interacted with three representations of a data set relating to outdoor pollution levels in six cities worldwide. The data representations consisted of a printed bar chart mounted to a wall, requiring only visual interaction, and two physical artefacts, named *DataBox* and *SonicData*, which required more active interaction. DataBox, a cube whereby each face represented a city, could be picked up and presented to a station that read an RFID tag embedded in each face. Once the country whose data is to be rendered is identified, the box, with the aid of an internal solenoid, begins to knock on its external walls. The air pollution data was thereby encoded as the rate of knocking, with a higher rate denoting a higher level of pollution. SonicData, on the other hand played sonic tones at frequencies mapped to the data set, with higher pollution levels mapped to higher frequencies. To activate a sonic tone participants interacted with a tactile interface, a small wooden cube moved over a labeled surface.

The study found that participants' affective responses to the data varied with representation modality. To allow for an analysis of the participants' responses, Hogan and Hornecker used bipolar constructs (based on Personal Construct Theory), with affective qualities, that were drawn from a focus group with the participants. In particular they looked at the Instinctual-Cerebral construct. Participants placed DataBox and SonicData at the instinctual end of the construct as interaction with these required the use of instinct to develop an understanding of what the representations meant. In contrast, the bar chart representation was categorised as cerebral, owing to the fact that that this representation is something that is learned and was viewed by participants as more of a tool.

The different modalities can be seen as resonating with the participants at different levels on an instinctual-cerebral scale. For example, the sound produced by Sonic Data, particularly those high pitch sounds associated with higher pollution, had a more personal and immediate impact on participants. This was because the sound was directly affecting them, being obtrusive and invasive in some cases and passive in others. Depending on the frequency of the sound, they either had a calming experience or a painful or annoying one. Here the data maps to a sensation that can be felt at a fundamental level, making it highly emotive and relatable and allowing, in this case, for a certain amount of empathy. With a physicalisation like DataBox, the interaction has less potential to be invasive while still allowing for an engaging and affective experience. In contrast to the two aforementioned data artefacts, the bar chart was emotionally divorced from the facts and constituted the most detached experience from an affective point of view.

An interesting point to note about the study is that different modalities led to different discussions about the same data set, potentially allowing the experiences to complement and enhance each other. Conversations among participant groups ranged from imagining what it must be like to live in a city with high air pollution and the consequent negative effects, generated by engaging with the DataBox and SonicData artefacts, to more speculative and analytical discussions about the causes of air pollution, which were fostered by interaction with the bar chart representation. Hogan and Hornecker [16] do note some shortcomings with the study, namely issues with concurrent perception when using the DataBox and SonicData artefacts arising from the temporal nature of the data rendering and a certain level of ambiguity in terms of some of the data mappings with the said-same artefacts. However such limitations could be overcome if the data were to be presented in a complementary manner.

It is therefore evident that the very way the same data is encoded can serve a multitude of purposes, from effectively and accurately denoting the quantifiable properties of data to eliciting an emotional response to the data in question. Rather than adopting one approach to encoding over another, they should, where possible, be used in tandem—either manifested in a single artefact or presented in a complementary manner, with the use of appropriate and considered transitions and juxtapositions. Physical, visual, and auditory encodings, could provide a multifaceted view on a given set of data, yielding data insights and understanding at the cognitive and affective levels.

If data is presented in familiar and typical formats alone, be those representations physicalised or otherwise, the capacity for developing unique, personal and unexpected insights, those ones that were not trained for, may be dulled somewhat. Such personal and unexpected insights arise from having engaged deeply with the data, seeing it from multiple perspectives, perhaps sometimes unfamiliar, uncomfortable and provocative, that challenge preconceptions and ultimately contribute to a richer understanding of the data in question. This is not to rule out the inclusion of traditional formats, such as bar charts and scatterplots, which serve an important purpose and should be a core part of any designed experience with a data set, rather the information clearly represented by such formats can be imbued with meaning and seen afresh via interaction with multiple representations.

2.6 Microsoft HoloLens

Given the possibilites offered by mixed reality headsets in terms of their ability to render and situate holograms in the physical environment with convincing realism, we noted that an effective data physicalisation platform could be developed if one could augment mixed reality holograms with haptic rendering. This, in turn, led to the idea of our tangible hologram system whereby holograms, generated by a Microsoft HoloLens headset, would be augmented with physical and material properties such as volume, weight, texture and inertia, thereby allowing them to be haptically sensed by users. In conferring these properties on holograms, their perceived *realness* is further enhanced, as making them touchable helps to maintain the illusion of embodiment so convincingly generated by the holographic headset. As previously mentioned, the benefits for augmenting holograms with haptic rendering goes beyond simply aligning visual and tactile perception to maintain the perceived congruence discussed by Jansen [21]. Rather, the main benefit, in terms of creating a system designed for the active exploration and analysis of data sets and, is to fully utilise that innate and powerful haptic perceptual channel to encode data in ways that allow for effective and affective interactions.

Figure 2.5: Microsoft HoloLens © Microsoft

One of the main advantages of using a mixed reality headset as a key part of the system is that its capabilities are not bound by many of the physical constraints pertaining to other technologies being applied to the problem of data physicalisation. The HoloLens is capable of generating holograms at any scale, be it a hologram that fits on the palm of your hand or one taking up a whole room. a given hologram may have a dynamic scale, transitioning from a particle the size of a dust mote to one dwarfing a human user, with such transistions taking place at whatever rate one deems appropriate.

Thus the holograms are instantly reconfigurable, satisfying one of those key properties of programmable matter. In addition, holograms have desirable properties that may prove more difficult to realise and refine in the short to medium term when one considers current technology and its rate of development. For example, a hologram may be one bound to the physics of the real world, thereby obeying the laws of gravity, or it may be capable of mid-air suspension and movement. In one instance, a rolling hologram might fall off the edge of a physical surface should it meet one and in another, maybe when the state of the hologram has been altered, it might continue rolling along the same plane in mid air, thus bending the rules of the physical world when required. Developing enabling technologies for materials and artefacts that exhibit such properties and dynamism is no small undertaking. While impressive technologies exist that have begun the process of realising material with anti-gravity capabilites, these are still largely in their infancy and either require substantial supporting infrastructure to make them work [29] or work with objects on the millimetre scale [41]. Consequently a significant amount of further development and refinement is required to replicate the qualities holograms can now already demonstrate.

In terms of data-driven environments, holograms can provide for the type of immersive analytics discussed by Cordeil et al. [7], where one can *"immerse themselves in their data in a way that supports real-world analytics tasks"* and undertake *walk-throughs* of a given data set. The data-driven environments envisioned here will be *"usable by experts and analysts to help in the detailed analysis of complex, big data sets"* while also being accessible to *"decision makers [and] the everyday public"*. Cordeil et al. note that the development of devices such as the Oculus Rift and Microsoft HoloLens can provide *"engaging and immersive experiences for a fraction of the cost of a CAVE"* [7], thereby increasing the viability of immersive analytics solutions and, consequently, the likelihood of their adoption by mainstream users. The ability to collaborate when exploring and analysing data, another important factor for those wishing to design interactive experiences with data physicalisations or immersive analytics environments, is facilitated through the HoloLens device and collaborators can be either collocated or based in remote locations. In any case, those collaborating on a project can share a data physicalisation or mutually explore a data-driven environment, thereby leveraging the benefits of shared experiences.

The Microsoft HoloLens offers more to the tangible hologram system than just the in-situ rendering of holograms. It also offers a range of input and output modalities, outside of the proposed haptic interface to be discussed later, and these modalities can play an important role in creating engaging and effective experiences with data. In terms of sound, for example, the HoloLens comes complete with an *audio engine* that provides *"the aural component of the mixed reality experience by simulating 3D sound using direction, distance, and environmental simulations"*[2]. Having such a capability as an intrinsic part of the tangible hologram system comes at no extra expense and may prove highly beneficial in helping to authenticate the perceived realness of rendered data physicalisations. For example, a hologram representing a physicalisation with an internal cavity may emanate a hollow sound when tapped, or holograms within a large data-driven environment may echo when interacted with. Directional sound may also play a very important role in navigation within data-driven environments alerting users that new representations have been manifested somewhere within the environment, signifying state changes to existing entities and providing cues to help locate relevant data.

Mid-air gestures and voice commands can also be used for interacting with the HoloLens, again adding to the advantages of using the headset in the tangible hologram system. Jansen et al. [23] note that when it comes to interacting with data physicalisations, *"recognizability and discoverability of interactions are important"*, while also noting that *"not all interaction styles can be easily expressed with [physical] affordances"*, citing symbolic and mid-air gestures as examples. Exploiting the gesture recognition capabilities of the HoloLens, then, one can ameliorate this situation, allowing for the provision of affordances outside of those directly manifested in the form of the physicalisation. symbolic and/or mid-air gestures can be easily differentiated from those interactions associated with the physicalisations. In this way, gesture-based interactions for higher-level operations such as selecting, ordering, juxtaposing and transitioning between different representations of the same data set, for example, can be separated from gestures for operating directly on physicalisations, such as deleting them.

While the HoloLens brings many advantages, the device does not come without its limitations. In the first instance, as the headset is a stand-alone computer, the benefits of mobility are offset by limited computing power, memory and storage. Being untethered from a computer means you can not access the processing power and other resources of a high-end desktop computer. In terms of resolution, the maximum supported resolution of the HoloLens is 720p (1268x720). In marked contrast, the 2.5K (2432x1366) resolution offered by the tethered Meta2 MR headset allows it to boast *"pho-*

[2]`https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_sound`

Figure 2.6: Meta2 AR headset © Meta Company

torealistic" holograms[3]. In developing HoloLens applications on the Unity[4] platform, it is recommended that the *Fastest* graphical quality setting is chosen for rendering, so that performance may be maximised. As the quality and resolution at which holograms are rendered affects their perceived realism, such limitations can impact the mixed reality experience on the HoloLens.

The device's field of view (FOV), in terms of what the user experiences, is also relatively restricted and an estimated 30°×17.5°[5] FOV provides for a view quite different to that which has been presented in HoloLens marketing material. In contrast, the Meta2 system, which is designed to work with holograms rendered within arm's reach, sports a 90° FOV, allowing for more compelling close up interactions with holograms. The fact that the Meta2 headset is designed to work with holograms close up, where one interacts with them directly in a more natural hands-on fashion, means that the Meta2 has some desirable capabilities when one considers the tangible hologram system. For example, the Meta2 occludes the user's hands as they interact with holograms and this is important in maintaining a sense of realism. It becomes even more important when attempting to make these holographic objects touchable. The HoloLens, on the other hand, does not provide hand-occlusion by default.

A pertinent question to ask at this point, given that the Meta2 headset would seem to have many advantages over the HoloLens in terms of the tangible hologram system, is why not choose this device over the HoloLens? One of the main reasons for selecting the HoloLens as the mixed reality compo-

[3] `https://buy.metavision.com`
[4] `https://unity3d.com`
[5] urlhttp://doc-ok.org/?p=1223

nent for the system is the issue of mobility. As already noted, the HoloLens's capacity for mobility brings with it its own drawbacks, yet the ability to freely move around within physical-digital environments and interact with and transition between them, untethered to a computer, far outweighs the aforementioned associated costs. Another factor in choosing the HoloLens is that one can begin to explore developing for the device without having to invest in it upfront. As there is no separate SDK for HoloLens development, Visual Studio with the Windows 10 SDK is used instead[6], and this is freely available to download. In addition, Microsoft provides several useful introductory videos through its HoloLens Academy portal to prepare and instruct those wishing to develop applications for the platform, while the applications themselves can be tested using the HoloLens emulator, again freely available from Microsoft. This, at least, allows for some form of evaluation of the platform before fully committing to it. The Meta2 SDK, on the other hand, is only accessible to those who buy the Meta2 Developer kit and while there are a number of Meta2 video tutorials online, it is still harder to evaluate. In both cases the Unity game engine is used to develop the mixed reality applications, allowing experience with the engine to transcend these platforms. The Meta2 offers the promise of interacting with holograms through direct manipulation with the hands, occluding the hands in the process, and this is one of its strongest advantages in terms of applicability to the tangible hologram system. However, several reviews of the system note tracking issues, b[hyphens]oth in terms of hand and inside-out positional and rotational tracking[7,8], and *'judder'* [9], which is a term for the uneven motion of holograms. Many reviewers cite the tracking achieved by the HoloLens as gold-standard, thereby helping to make the case for its adoption over the Meta2.

One final consideration in terms of developing holographic applications where holograms are designed to be viewed and interacted with at close range is the problem of vergence-accommodation conflict, an issue that both manufacturers of VR and MR/AR systems are currently trying to deal with. In discussing vergence-accommodation conflict, Hoffman et al. [15] note that the *"uncoupling of vergence and accommodation required by 3D displays"*, *"frequently reduces one's ability to fuse the binocular stimulus and causes*

[6]`https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools`

[7]`https://www.roadtovr.com/meta-2-development-kit-hands-on-could-do-for-augmented-reality-what-oculus-rift-dk1-did-for-virtual-reality`

[8]`https://uploadvr.com/meta-2-hands-ar-svvr`

[9]`https://www.theverge.com/ces/2017/1/6/14187780/meta-2-augmented-mixed-reality-headset-hands-on-ces-2017`

discomfort and fatigue for the viewer". To help reduce the effects of vergence-accommodation conflict, Microsoft recommends that holograms be placed in an optimal zone of between 1.25 and 5 metres, noting that *"Discomfort from the vergence-accommodation conflict can be avoided or minimized by keeping content that users converge to as close to 2.0m as possible (i.e. in a scene with lots of depth place the areas of interest near 2.0m when possible)"*[10]. The vergence-accommodation conflict is an issue inherent in any 3D display, be it VR or MR based, and any system working with such a display has to assume its presence and cope with the problems associated with it.

[10] https://developer.microsoft.com/en-us/windows/mixed-reality/hologram_stability

3
Related Work

In selecting a suitable approach towards the problem of haptic rendering, one needs to consider a system that best exploits the possibilities offered by the HoloLens headset and the functional requirements of such a system. As previously mentioned, one strong advantage of the HoloLens is the capacity for untethered mobility. To fully exploit this advantage, then, the system as a whole should be mobile, and, given that the HoloLens is wearable, it makes sense that the system too should be wearable. The criterion for mobility rules out those approaches currently limited by the need for significant supporting infrastructure. Were it possible to have a system such as TRANSFORM [19] at a scale where it might be feasible to house it in an actuated wearable device, such as a small platform with many degrees of freedom, then this, indeed, may prove to be an ideal solution. However, as it stands, the powerful capacity for such shape display devices to dynamically render form and provide for bidirectional input/output is offset by a distinct lack of mobility.

Yet mobility is not the sole determining criterion. One must also take into account how generalised or specialised the device needs to be, in addition to the range of exploratory procedures one would want to support through the solution. Ideally, an effective platform for building data physicalisation approximations would be general enough to allow for rendering a sizeable range of physical/material properties, including weight, volume, inertia and texture. Therefore, the solution should be a platform generalised

Figure 3.1: Tactai Touch
© Tactai

Figure 3.2: CyberGrasp glove
© CyberGlove Systems

enough to support all those exploratory procedures so important in haptic perception. Considering these factors when looking at options that offer mobility thus helps to eliminate further lines of enquiry. Take, for example, the Tactai Touch device[1]. This small and highly innovative piece of hardware can be worn on one finger, or multiple units can be worn on multiple fingers, and can render *"life-like touch sensations with precision engineering, and high-resolution hand-and-finger tracking to enable unprecedented levels of immersive control and interaction in touch driven applications"*[1]. In terms of augmenting holograms with haptic rendering, this device would appear to have many advantages—and it certainly does.

Tactai Touch is based work undertaken by Kuchenbecker et al. [25], whereby a small cup-like device, the *"Touch Thimble"*, was attached to the end of a grounded manipulandum, in this case the Phantom SensAble, and allowed users to sense virtual surfaces with their fingertip. Tactai Touch, on the other hand, is not designed to attach to any other device and is capable of providing rendering through an internal actuated vibrating platform. The benefits of using such a small mobile device in VR/MR applications are many. However, if one was to employ the Tactai Touch alone for providing haptic rendering, the number of exploratory procedures supported would be limited. There would be no provision for unsupported holding to determine a hologram's weight or for enclosing a hologram with one's hand to determine its volume and global form, as there would be no force acting against the

[1] `http://www.tactai.com`

hand as a whole. Additionally, one could not apply pressure to a hologram to determine its relative hardness.

Force-feedback gloves, in contrast, provide forces that act against the fingers of the hand, thereby allowing one to sense the form, volume and stiffness of a virtual object through exploratory procedures such as enclosure and pressure. To achieve those resistive forces applied to the fingers, gloves such as the CyberGrasp[2] and the Rutgers Master II [4] employ a hand-worn exoskeleton whereby multiple fingers are acted upon individually through the use of seperate actuators. While force-feedback gloves can support enclosure and the application of pressure, and still offer a high degree of mobility, they do have a number of drawbacks. Burdea and Coiffet [6], in discussing some of the limitations of the CyberGrasp glove, note that the weight of the part of the system that must be worn on the arm, some 539 grams, can lead to user fatigue. They also note that the glove has an *"inability to simulate the weight and inertia of the grasped virtual object"*, although the glove can be coupled to a grounded mechanical arm, the so-called CyberForce device, to simulate these properties. Coupling to this extra device comes at the expense of mobility, however. In addition, the cost of these devices, given their complexity and the level of engineering required to produce them, is prohibitive. The CyberGrasp glove, for example, can retail at several tens of thousands of dollars [6]. This is in marked contrast to the $12 cost for developing Tactai Touch prototypes[3].

In terms of supporting the sensing of weight and inertia, using a grounded articulated device is a well-established and often employed strategy. One popular grounded device used in applications requiring haptic rendering is Geomagic's PHANTOM model. The PHANTOM device [31], a small mechanical arm designed to sit on a desktop, was one of the earliest of these models. Here users insert their finger into a thimble mounted to the end of the manipulandum, with the device providing *"a force-reflecting interface between a [human user's fingertip] and a computer"*. The PHANTOM can also be coupled with a stylus for interactions requiring a higher degree of precision. As with the other force-feedback options mentioned, there is a trade-off between what exploratory procedures a device can support and the restrictions incurred in order to provide this support. With devices like the PHANTOM, the ability to augment virtual objects with a sense of weight and inertia requires the device to be grounded. This is acceptable and even preferred for many applications. However, for TangHo mobility is an important factor thus working with grounded devices needs to be ruled out.

[2] `http://www.cyberglovesystems.com/cybergrasp`
[3] `http://www.tactai.com`

Figure 3.3: PHANTOM device (now TouchX) © Geomagic

Figure 3.4: Snake Charmer © Araujo et al.

One approach for haptically augmenting virtual objects that sits quite close to our tangible hologram solution presented in this thesis is the work of Araujo et al. [1] and their *'Snake Charmer'* device. This is, in effect, a robotic arm possessing of many of the capabilites required by the tangible hologram system. The arm itself, through actuation, can position an end-effector at the point of user contact with a virtual object, and, in turn, can be used for user input, allowing one to move a virtual object by manipulating the end effector of the arm. Another important feature of Snake Charmer is that it is able to simulate moving content. In addition the robot arm is capable of selecting and attaching a range of end-effectors facilitating different means of physically sensing a virtual object, such as through temperature, pressure sensing and via various textures, as requried by the application. As noted by Araujo et al. [1], Snake Charmer overcomes some shortcomings associated with the PHANTOM device in that a user does not have to constantly hold the device in their hands to get haptic feedback. Instead, Snake Charmer provides hands-free haptic rendering, freeing up the user's hands and affording a more natural form of interaction with physically augmented virtual content. In developing the Snake Charmer, Araujo et al. work within McNeely's Robotic Graphics paradigm, where *"force feedback is provided by interactions between the human operator's body and specialized external (as opposed to worn) robots"* [32]. Indeed, this quote from NcNeely serves to highlight one of the key differences between the tangible holograms system and the Snake Charmer solution—the wearability and mobility component.

While the Snake Charmer solution was shown to work with a VR headset, there is no reason why it could not be adapted to work in a similar way with an MR headset such as the HoloLens. With this in mind then, one key difference between the solution developed by Araujo et al. [1] and the work presented in this thesis is the issue of mobility. The tangible hologram system is designed to be worn, therefore the device is grounded with respect to the user themselves. As such, sensations such as weight and inertia can be rendered without the need for grounding to a static point, thus achieving the benefits of grounding the device without incurring the cost of losing mobility. That being said, the Snake Charmer device is a highly innovative piece of hardware possessing many desirable capabilities.

4 TangHo

Initial research in the field of data physicalisation and work relating to it, in addition to the work of Ishii and others at MIT Media Lab's Tangible Media Group, led to an exploration of the possibility and feasibility of designing and developing a new plaform for further exploring the possiblities of data physicalisation. The proposed solution would be an system in the tradition of those developed by researchers at the Tangible Media Group, seeking to approximate the programmable matter described in Radical Atoms [18] by providing or emulating some of its capabilites. As previously mentioned, in discussing the opportunities and challenges for data physicalisation, Jansen et al. [23] note the important role programmable matter, and current technologies working towards the promise of such a dynamic material, may play in the field of data physicalisation. Indeed, the capabilities afforded by exisiting systems such as the TRANSFORM [19] and ChainFORM [34] shape changing interfaces already provide a suitable and versatile platform for designing various forms of data physicalisation and interactions therewith.

4.1 The Tangible Hologram System

The tangible hologram system seeks to work in a complementary manner with current and developing technologies that may one day play a significant role in achieving programmable matter. In setting out the initial idea for a tangible hologram system, Signer and Curtin [42] note that *"existing solutions start with the physical object and try to make the physical material and interfaces more configurable in order to enable dynamic physical affordances and support dynamic data physicalisation"*. In contrast, they describe the approach taken by a tangible hologram system as one that starts with holograms *"that are perfectly embedded in physical environments"* and *"rather than making physical objects more configurable, [the] challenge is ... to add physical features to the already perfectly configurable digital holograms"*. In realising the system, we aim to approximate the full potential of programmable matter and by doing so through a combination of existing technologies, we hope to place this dynamic material in the hands of interaction designers and technologists today rather than tomorrow. These groups can then move beyond the storyboard and begin developing real experiences in a rapid and inexpensive way, unbound by technical limitations. The system may help to overcome the *inherent complex workflows* involved in prototyping and building data physicalisations [23], serving to provide a high fidelity proxy for a physicalisation so that its effectiveness in affording appropriate interactions and knowledge transfer may be tested. Such interactions may lead to new directions for developing actual technologies to materialise a promising idea.

There are two main components in a tangible hologram system: the headset required to generate the holograms and the haptic system needed to augment these holograms with physical properties. As noted by Signer and Curtin [42], the headset should offer the capacity to *"track the spatial layout of the environment as well as any physical objects via depth and environmental cameras"*, in addition to the actual rendering of holographic content. In this regard, the previous discussion on the capabilities of the Microsoft HoloLens and its advantages over other options serves to elucidate the rationale behind the decision to adopt the device as the tangible hologram system's mixed reality component. In terms of the system's haptic component, the related work section saw the analysis of a number of current systems offering haptic interfaces to virtual content. During this analysis, it was noted that mobility and the need to support a wide range of exploratory procedures were essential requirements to a tangible hologram system. These requirements helped to rule out a great many of these haptic rendering approaches as, while they represented highly innovative and able solutions in their own right, they did not entirely satisfy the needs of the system. Ultimately the choice of wearable robotic arms to provide the physical augmentation of holographic content proved to be the most suitable and effective option.

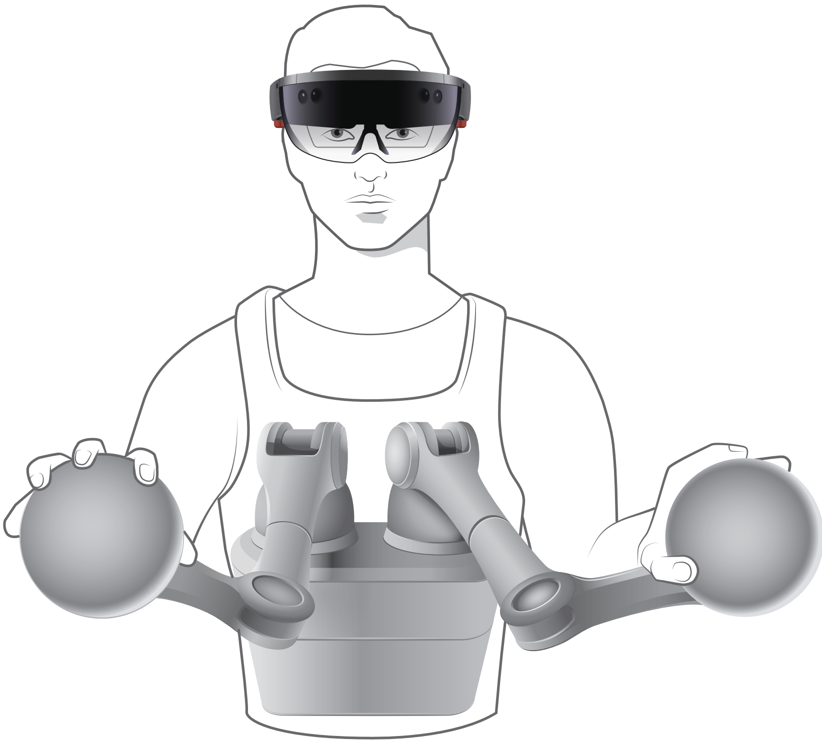
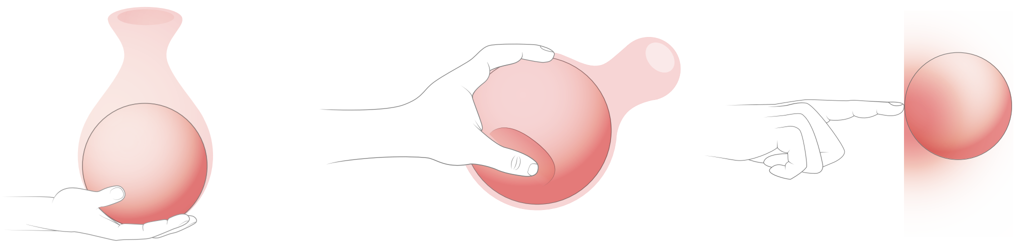


Figure 4.1: Concept sketch of the tangible hologram system

Figure 4.1 shows a conceptual drawing of how the system would look when worn by the user. Two sphere-shaped end-effectors sit at the end of two robotic arms, both of which are anchored to a base unit worn by the user. Thus, the arms are grounded devices with respect to the user yet still mobile in that the user is unrestricted in terms of their mobility. As the user sees holograms through the headset they may haptically interact with them through the end-effectors on the robotic arms and there are a number of different interactions types envisaged. In the first instance, the system should permit a user to reach out and touch a given hologram. To do this the system must move the end-effector of the appropriate arm to a Cartesian pose in physical space such that the surface of the end-effector represents the surface of the hologram in its virtual space. In this way the system haptically augments the hologram only at the point of touch, thereby allowing it to provide haptic rendering for arbitrarily-shaped holograms at any scale. Ideally the system should feel responsive and exhibit relatively little lag in performing the end-effector positioning. Nielsen [35] notes that a reaction time of 100ms is roughly the limit for giving a user the impression of instant system responsiveness. This limit bounds the full cycle of tracking a number of different elements, such as recording the positions of the end-effectors and the user's hands, transmitting the various data and performing the necessary computations on that data. Sensor information about the position and orientation of robotic arm limbs, and the joint angles derived from therefrom, must also be factored into the loop. Ensuring that this cycle can complete within the 100ms bound means that the system must be as optimal as possible. One clear approach in helping to attain optimal performance is to ensure that complex calculations are performed where computational resources are freely available. Additionally, the system needs to predict which virtual object a user wishes to interact with and begin the process of positioning an end-effector before the point of contact between a user's hand and a given hologram is reached. The end-effector should be present at the point of contact at the moment the user's hand or fingers reach it.



(a) Holding (b) Squeezing (c) Touching

Figure 4.2: Exploratory procedures with a sphere-shaped end-effector

The sphere end-effectors shown in the concept drawing of Figure 4.1 are simple, yet they support a number of exploratory procedures, such as those illustrated in Figure 4.2. For example, unsupported holding, shown in Figure 4.2a, enables a user to determine the weight of a hologram. This can be controlled by exerting the appropriate amount of relative directional force on the users hand when the system detects that the user wishes to evaluate the virtual object's weight. If the object is completely inert then the end-effector can be positioned at the point of contact with the hologram and simply lock in place. Holographic artefacts with differing mass can have a factor assigned to them so that the necessary amount of opposing force may be calculated. A similar approach may be employed when a user pokes or touches an object with varying amounts of applied force. Figure 4.2b shows how a sensation of squeezing a virtual object may be rendered through an end-effector made of a soft elastic material. As a user squeezes the physical end-effector, a deformable hologram, that retains a consistent volume, reacts accordingly. The detection of the squeezing action could be achieved in multiple ways. For example, small sensors could be distributed evenly throughout the sphere and, as their relative positions change, the recorded changes could be used to update the appearance of the hologram.

The interactions depicted in Figure 4.2 can be seen to represent both the sensing of holographic objects and their direct manipulation through the interface—if one considers that the actions depicted in Figure 4.2a and Figure 4.2c may have an effect on the position of the holograms. In any case, an important consideration with such interactions is that the system should provide some form of hand occlusion to help maintain the perceived authenticity of the interaction. As previously mentioned, the HoloLens does not provide this capability by default, although it is conceivable that this feature may be added in future iterations. In addition to this, the hand and finger tracking capabilities of the headset are not entirely accessible. One can access the moving position of the center of the hand when a known or custom gesture is recognised, however, at present, it is not possible to track the hands and fingers as one would with a device like Leap Motion[1]. Again, hand and finger tracking may become available in future iterations. As it stands, however, these limitations constitute an unfortunate drawback in terms of employing the device as part of tangible hologram system, detracting a little from all the potential benefits the device offers to the system. It may be possible, in the interim, to harness the capabilities of a hand tracking solution by incorporating it into the system but this undertaking can be regarded as future work.

A more involved interaction is shown in Figure 4.3, where a user-centric view is depicted. Here a user interacts with a data set represented as a graph, which is projected into the surrounding environment. The user can interact with individual nodes by manipulating the end-effectors, rearranging and repositioning them as needed. As the user moves around the graph the spherical end-effectors continually reposition themselves to align their surfaces with the surfaces of those nodes closest to the user. As noted by Signer and Curtin [42], who present this scenario in an earlier work, *"the spheres can not only be used as input devices but the robotic arms may also apply some directional force to the spheres in order to provide some additional computer-generated haptic force feedback based on the underlying digital model"*. Indeed, the end-effectors may also *"provide non-visual supplemental feedback about the underlying digital model or data via physical variables such as shape, texture or temperature"* [42]. Adding the capacity for the tangible hologram system to render these physical variables will require the development of different end-effectors and it is hoped that these may be realised through future work on the system.

[1] `https://www.leapmotion.com`

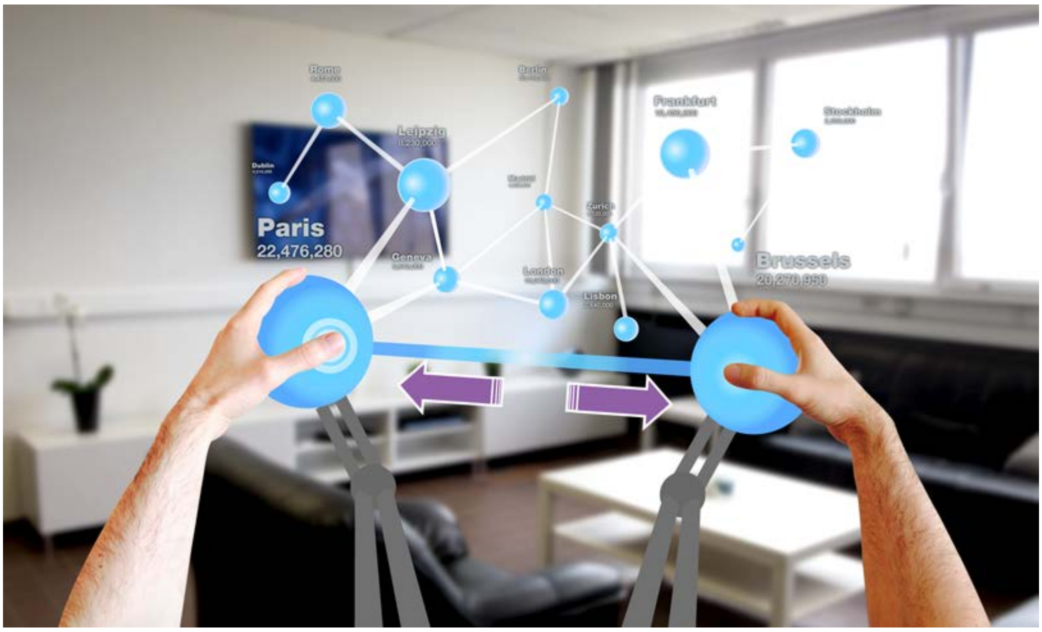


Figure 4.3: Tangible holograms user view mock-up

As is evident from the descriptions of the aforementioned interactions, an alignment of physical and virtual coordinate systems is essential so that the tangible hologram system can correctly position and rotate the end-effectors into the poses required to achieve congruency between their surfaces and those of touched holograms. In effect, the physical position and orientation of the robot arms' end-effectors need to be tracked and translated into virtual coordinate system of the holographic environment. This is in addition to the required hand tracking, which was previously discussed. An important question at this point is which approach to tracking best fits the requirements of the system, especially with regard to the concerns around mobility. A number of highly effective and commonly employed motion capture solutions exist, with the OptiTrack[2] being among the most prominent. With the OptiTrack solution, high precision cameras and photo-reflective capture markers are used to provide accurate tracking within a given area. One disadvantage to motion capture solutions such as this, in terms of the tangible hologram system, is the need for extra hardware and the fact that the tracking system needs to be set up in every physical environment you wish to interact with. An ideal solution would be to leverage the numerous sensors present in the holographic headset to provide the necessary tracking instead. In this way, the approach helps to maintain the mobility of the system, whereby a user

[2]`https://optitrack.com`

can freely move between environments and even work in dynamic outdoor environments without the need to set up extra tracking infrastructure.

The concept drawing in Figure 4.1 shows the tangible hologram system as it is to be worn by the user. As can be seen from the illustration, two robotic arms sit on a slightly tilted platform that, in turn, rests on a supporting structure braced against the users lower body. The platform is slightly tilted to situate the arms back a little from the user, thereby preventing them from being too close, while still allowing the supporting structure to have a relatively compact form factor. Ideally the tilt angle of the platform would be adjustable, granting a certain amount of flexibility. However the tilt should be restricted to a range that still allows for the effective movements of the arms. Should the tilt be too vertical, the resultant forces acting on each base joint, produced by the uneven distribution of each arm's weight, would prevent smooth arm movements at this joint. The haptic unit itself should be as lightweight as possible to reduce strain and user fatigue when used for extended periods of time. In this regard, composing the unit from 3D-printed plastic components would help to reduce weight while still offering a high degree of robustness. To make the wearing of the unit as comfortable as possible, a vest or harness would help distribute the weight of the system and reduce the impact of sudden arm movements on the user experience.

Each robotic arm would need to have six degrees of freedom (6DOF), such that the arm can move freely across and about three perpendicular axes thus giving three translational and three rotational degrees of freedom respectively. Such freedom is required if the arm is to be capable of positioning the end-effector at any point and orientation in 3D space. The end-effectors shown in the illustration are simple spheres and so their orientation may not seem too important a concern. However, it is envisaged that, with time and further development, a range of end-effectors would be developed, with each one affording a different form of haptic rendering. The orientation of such end-effectors in 3D space may indeed become an important factor.

On the computational side, kinematic solvers constitute an important requirement. Forward kinematic calculations determine the pose of the end-effector, and the arm as a whole, when supplied with known joint angles. Inverse kinematics, on the other hand, calculate the joint angles required to achieve a desired end-effector pose, which is supplied to the solver. The tangible hologram system requires both forms of kinematic solvers to carry out its functionalities effectively. An important input to both forward an inverse kinematic equations are the current joint angles of the manipulator, thus placing a requirement on the system to sense and record these angles. Were the haptic system to be mounted to a stationary point such as a desktop or table then it might be feasible to employ a process of dead reckoning to keep track of the arms' current joint angles. The process would entail knowing the starting angles of the arms before any user-determined actuation of the arms takes place. This could be easily achieved by calibrating the joints, by running them to physical hard limits, and then setting the arms to a known *home* configuration or pose. Each subsequent movement will then update the current configuration, providing a good estimate of each joint angle.

The tangible hologram system is designed to be worn, however, ruling out the possibility of using a convenient process such as dead reckoning. The pose of the haptic system's base and, consequently, the robotic arms, is in a constant state of flux. What is required instead, then, is the real-time capture of the orientation of the base and each of the limbs of the robotic arms. With this information it will be possible to derive the relative joint angles necessary for the kinematic calculations. The position of the base relative to the virtual coordinate system of the holographic environment is also required and can easily be derived from the end-effector position recorded from the headset once the relative joint angles are known. To detect the real-time orientation of the base and limbs, Inertial Measurement Unit (IMU) sensors will need to be availed of. The IMU sensors required for the tangible hologram system will need to be composed of triaxial accelerometers, gyroscopes and magnetometers so that roll, pitch and yaw angles may be derived. Another important factor to consider then working with a worn haptic system is that the user themselves contribute movement to the system. To expand on this further, imagine the case where a user wishes to interact with a given hologram and the system detects this intention and begins the process of moving the end effector to the desired location. As the arm begins to move, the user too may turn a little towards the hologram, thus contributing some of the movement required to reach the end goal. If the system does not take this into account the arm may well overshoot or, indeed, undershoot the target, leading to an undesired outcome. Ideally the system will need to continuously sample where an end-effector is at a given moment and update its calculations on the fly. This means a trajectory is made up of a number of discrete steps, with each one taking into account current tracking data. This places some challenging performance requirements on the system, given the aforementioned 100ms bound essential to demonstrate a responsive system.

4.2 Addressing Data Physicalisation Challenges Through the TangHo System

As discussed in earlier sections, an important function of the TangHo system is to facilitate the exploration of interactions with data physicalisations, thereby providing a platform upon which ideas for data physicalisation may be developed and evaluated. It is hoped that, through its capabilities, the tangible hologram system may help to overcome some of the limitations associated with technologies currently considered as applicable to the data physicalisation problem. In outlining the challenges for implementing data physicalisations, Jansen et al. [23] note the limitations of current fabrication technologies when trying to create physicalisations that can encode a wide range of physical variables. They also remark that any generated physical encodings should be immune to alteration resulting from age and repeated use. Achieving these goals may take several years to realise. In terms of *active physicalisations* and their role in helping to overcome some physicalisation implementation issues there are also limitations. For example, in discussing how magnetic fields and acoustic levitation may help physicalise free-floating objects, Jansen et al. note that *"fine control over 3D geometry is hard and no technology exists yet that can successfully physicalise data encodings as simple as 3D scatterplots"*. In lieu of such technology, the tangible hologram system might act as a suitable stand-in. It can render the effect and appearance of a physicalisation without any of the aforementioned fabrication issues and can display free floating forms while only needing to render haptic feedback the point of user contact with the representation.

In addition, the tangible hologram system is capable of handling those capabilities associated with dynamic physicalisations, such as transitioning between representations and data sets, ensuring that any changes made to the underlying model are manifested in the physicalisations and updating interfaces and physical affordances as required by the state of the application, the requirements of the user and the task at hand. Jansen et al. [23] also note that there are trade-offs between different enabling technologies. Whereas active physicalisations created from actuating technologies offer certain advantages over passive physicalisations, they cannot be duplicated with the ease of some passive physicalisations, removing the ability to arrange the physicalisations side be side for further analysis and exploration. This is an important capability and Hull and Willett [17], in exploring how architectural models may provide some inspiration for data physicalisation, discuss the importance of collections of working model iterations in providing a visual record of the architectural design process, allowing teams to reflect on the direction of concept development as it evolves over time. They go on to note that collections of working models also serve to situate different design options in the same space to allow for effective comparison. Again the tangible hologram system can facilitate such collections of physicalisations allowing them to retain their location in the mixed reality world between sessions and for any required period of time, owing to a capability of mixed reality headsets also recognised by Hull and Willett. In addition the system may be used as an authoring and archiving tool for physicalisations, allowing them to be materialised, whenever required, by sending virtual 3D representations to a 3D printer—thus giving them the *always on* property of real physicalisations.

The tangible hologram system thus overcomes many of the implementation challenges mentioned by Jansen et al. [23], without the need for, and associated limitations and cost of, sophisticated technologies. The system can provide immediate reconfigurability, offering the ability to transition between different resolutions of the same physicalisation or different physicalisations of the same data set with relative ease. Rendered physicalisations can be presented alongside dynamic 2D representations of the data allowing for the creation of blended environments that optimise interaction with and between 2D and 3D forms. In terms of interaction, both direct manipulation and mid-air gestures, as well as voice commands are supported, thus providing the capacity to assign appropriate forms of interaction to areas where they are most effective. Additionally, the tangible hologram system may prove very benefical in the context of collaborative data analysis and exploration through physicalisations, another challenge when working with physicalisations [23], achievable by leveraging the collaborative capabilities of a holographic headset such as the Microsoft HoloLens.

In terms of providing for more affective interactions with data physicalisations, similar to those discussed in the background section earlier, the tangible hologram system can offer much in the way of possibilities. For example, the tangible hologram system has the potential to render multiple representations of physicalised data in a unified and structured way, allowing for the design of purposeful and meaningful interactions with different modalities with coherent and seamless transitions between them. The system can also render auditory and visual information in addition to the physicalised data, augmenting physicalisations and interactions where needed. As such Tangible holograms can become a platform upon which richer, more engaging and more memorable experiences with data may be built.

As has just been discussed, the tangible hologram system may be quite beneficial in terms of exploring the possibilities of data physicalisation, helping to overcome current limitations restricting what can be developed and tested. This, in turn, may help to broaden the appeal of using data physicalisations in exploration and analysis tasks as the approach becomes more feasible. In developing a working system, however, one still needs to consider the limitations of individual components and the impact of these limitations on the whole solution. A case in point is the use of the Microsoft HoloLens as the system's mixed reality component. While the marketing material for the device depicts a certain view of how the user experiences holographic content, in reality, the holograms are rendered in a window with a relatively small FOV that sits at the centre of one's own field of view. Any holograms straddling the boundary of the window are truncated. Over time, one gets accustomed to this limitation and learns to work with it. However, the tangible holograms system is designed to augment holograms with haptic rendering, thereby making them touchable and such interaction requires a user to be within arm's reach of a hologram. In this context, issues associated with the limited field of view become more pronounced and holograms viewed at this close distance are more noticeably truncated. This example serves to highlight the difficulties in working with emerging technologies, sometimes in ways that were not entirely envisaged by their creators. It is important to note, however, that while the user experience may not be completely optimal at the moment, this will improve with further refinements of the underlying technologies and, although currently limited in some respects, present technologies are good enough to begin building effective prototypes to demonstrate what is possible.

5
Implementation

The implementation of the tangible hologram system, which saw the synthesis of a number of different technologies and concepts, can be divided into two broad areas, namely hardware and software. This section, therefore, is divided accordingly. The hardware section covers the construction of the TangHo system's robotic arms with Lego Mindstorms along with the design and fabrication of the haptic unit's base section. Issues concerning the effective use of Inertial Measurement Units (IMUs) to determine the orientation of certain system elements also relate to hardware and so are discussed here. The software section describes how those components running on various elements of the TangHo platform interoperate and details how data flows through the system as a whole. Kinematic solvers and the role they play in the system are also explained here, as is end-effector detection and tracking via the Microsoft HoloLens.

5.1 Hardware

The system hardware needs to meet several requirments in order to facilitate realising the tangible Hologram system. The system is mobile and thus needs to be made as light as possible while still retaining a high degree of robustness, given the demands and strains placed upon it during operation. In addition, the arms of the system need to be capable of actuating at high speed to help make the system as responsive as possible. As mentioned earlier, Nielsen [35] notes that a reaction time of 100ms is roughly the limit for giving a user the impression of instant system responsiveness, although he also notes that response times should be as fast as possible, without overwhelming the user. Ideally one or both arms should move into position just as the user is about to touch a virtual object, placing the end-effector in just the right position so as provide haptic rendering at the point of contact. However, the actuation of the arm is only the final stage in the system's haptic rendering cycle. Information needs to make its way through several system components before the arms have the information they need to begin actuating and the latencies associated with reading various sensors and sending and processing data demand that the arms of the system need to be capable of fast actuation if the system as a whole is to be perceived as suitably responsive. Ensuring that the components of the arms are as light as can be and trying to predict, as much as possible, when the arms might need to be actuated will also help in this regard. In order to render haptic feedback of virtual objects the arms of the system need to be capable of producing a certain amount of force, in addition to supporting and moving their own weight.

The ergonomics of the system require careful consideration and the device should be comfortable to wear, exerting as little strain as possible on the user, given that it may be worn for extensive periods of time. As such the weight of the system should also be distributed evenly. It is also envisaged that future developments of the Tangible Holograms system will see different end-effectors being designed for different purposes or applications. For example, different end-effectors may afford different rendering resolutions, with varying strategies being employed for doing so, or an application may require temperature as a physical variable for encoding data, so necessitating an end-effector with those capabilities. At any rate, the system should allow for a certain degree of customisability and, as such, should be made modular where it is viable to do so.

The processing hardware of the tangible hologram system is distributed across a number of different components, namely the Microsoft HoloLens, an Apple MacBook Pro and three Lego Mindstorms EV3 *bricks*[1] which are small computers running a Linux OS. Figure 5.1 gives an overview of the processing hardware specifications.

[1]`https://shop.lego.com/en-US/EV3-Intelligent-Brick-45500`

	Microsoft HoloLens	MacBook Pro	EV3 Brick
Main Processor	Intel Atom x5-Z8100 1.04 GHz Custom Holographic Processing Unit (HPU 1.0)	2.4 GHz Intel Core i7	TI Sitara AM1808 (ARM926EJ-S core) @300 MHz
Memory	2GB RAM 64GB Flash	16 GB 1600 MHz DDR3	64 MB RAM 16 MB Flash microSD Slot (max 32GB)
OS	Windows 10 (32 bit)	OS X 10.10.5	leJOS 0.9.1
WiFi	Wi-Fi 802.11ac	Wi-Fi 802.11 a/b/g/n	Optional via USB
Bluetooth	Bluetooth 4.1 LE	Bluetooth v2.1 + EDR	Bluetooth v2.1 + EDR

Figure 5.1: Processing hardware overview

5.1.1 Lego Mindstorms

Given the time limitations, the selection of a suitable means by which the system's arms could be iteratively developed into a working prototype, while providing the flexibility to learn from errors without too much associated cost, both in terms of time and resources, was important. This necessity pointed towards a modular solution where components could be interchanged and adapted as needed with relative ease. As such, the Lego Mindstorms system was initially chosen as it satisfied many of these requirements. In terms of prototyping the system's arms, Lego Mindstorms has proved to be a good choice and has demonstrated itself to be a suitably modifiable and robust system. It presents a unified interface through which to develop, with servo motors and sensors of various types being easily integrated into a given construction, thereby providing sensing and actuation. Indeed, Lego Mindstorms is used in many schools and universities around the world to teach robotics and many concepts are prototyped and tested utilising it. A testament to what is achievable with the Mindstorms system is the *CubeStormer 3* robot[2], capable of solving a Rubik's cube in 3.256 seconds, the current Guinness World Record for a robot. Another notable example is the near life-size ABB industrial robot built by Madsen and Lauesen for ABB Robotics[3]. In terms of building serial link manipulator robot arms with Lego Mindstorms, there are several highly refined examples online, in particular the work of Akiyuki[4], Odenthal[5] and OrangeApps GmbH[6,7], that provide good starting points from which to learn about designing and building such manipulators.

A Mindstorms system may be programmed in a number of different ways. The default method of programming is to use a GUI-based visual programming language based on National Instrument's LabVIEW software and, while it is possible to build sophisticated applications using this approach, there are many alternatives. The brick can be programmed in a range of different languages including Java, C, C++, C# and Python and this adds to the flexibility of the solution. The leJOS Java API was chosen to build those parts of the system running on the EV3 brick as it offers many important classes for working with the brick's hardware, for accessing and actuating the servo motors and also has drivers for a wide range of sensors, including third-party sensors. Another important factor in selecting the leJOS API was the large community of developers working with it, as well as a number of university programs teaching robotics, whose materials are freely accessible online. The leJOS Java API will be discussed again later in the software implementation section.

The servo motors used within the Mindstorms system bring both advantages and drawbacks. On the one hand they are capable of precise control, are accurate to within one degree and are easily integrated into a given construction. The tangible hologram system's arm prototype uses three large and three medium servo motors. The large servo motor runs at 160-170 rpm, with a running torque of 20 N/cm and a stall torque of 40 N/cm, while the medium servo motor runs at 240-250 rpm, with a running torque of 8 N/cm and a stall torque of 12 N/cm. As such they provide a reasonable amount of speed and torque. However, given the size of the arms required by the system, several gear trains had to be designed to ensure that the joints could move their associated limbs under their own weight. The design effort here proved more time consuming than anticipated as striking the balance be-

[2] `https://www.lego.com/en-us/mindstorms/news/2014/march/cubic-stormer`
[3] `https://www.youtube.com/watch?v=-tsPuHaHFDw`
[4] `http://akiyuki.jp`
[5] `https://www.flickr.com/photos/siouxnetontrack/sets/72157660706604268`
[6] `https://www.youtube.com/watch?v=ozfXcvSCNNQ`
[7] `https://www.youtube.com/watch?v=PWD32CspN70`

tween torque amplification, or mechanical advantage, and loss of speed, was non-trivial. Figure 5.2 shows a number of different trial gear configurations.

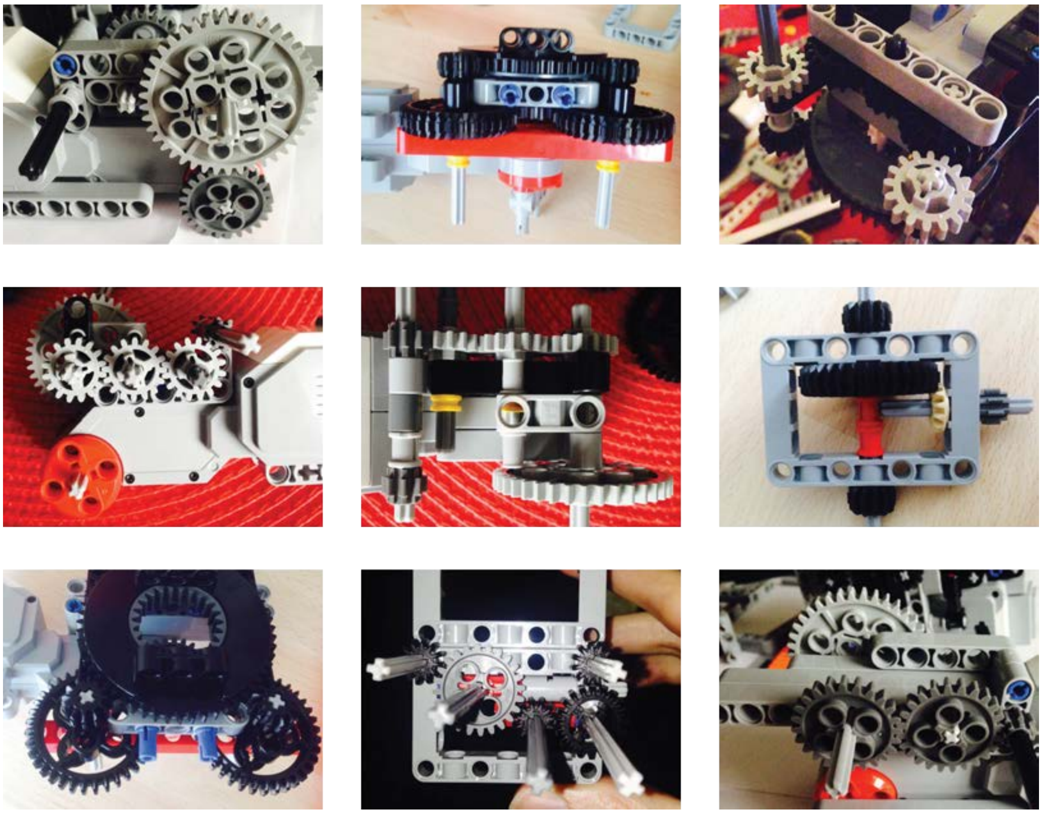


Figure 5.2: Trialling different gear configurations

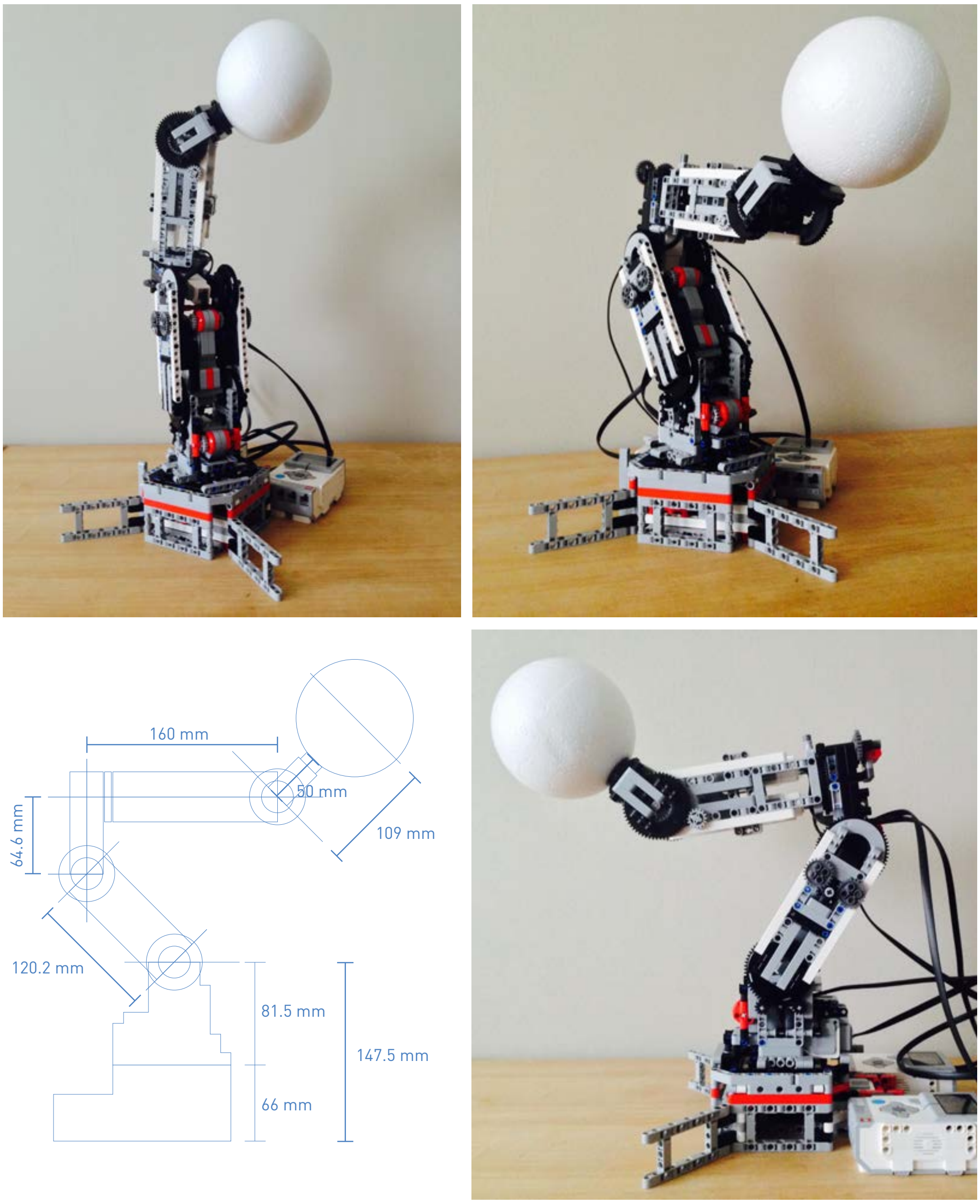
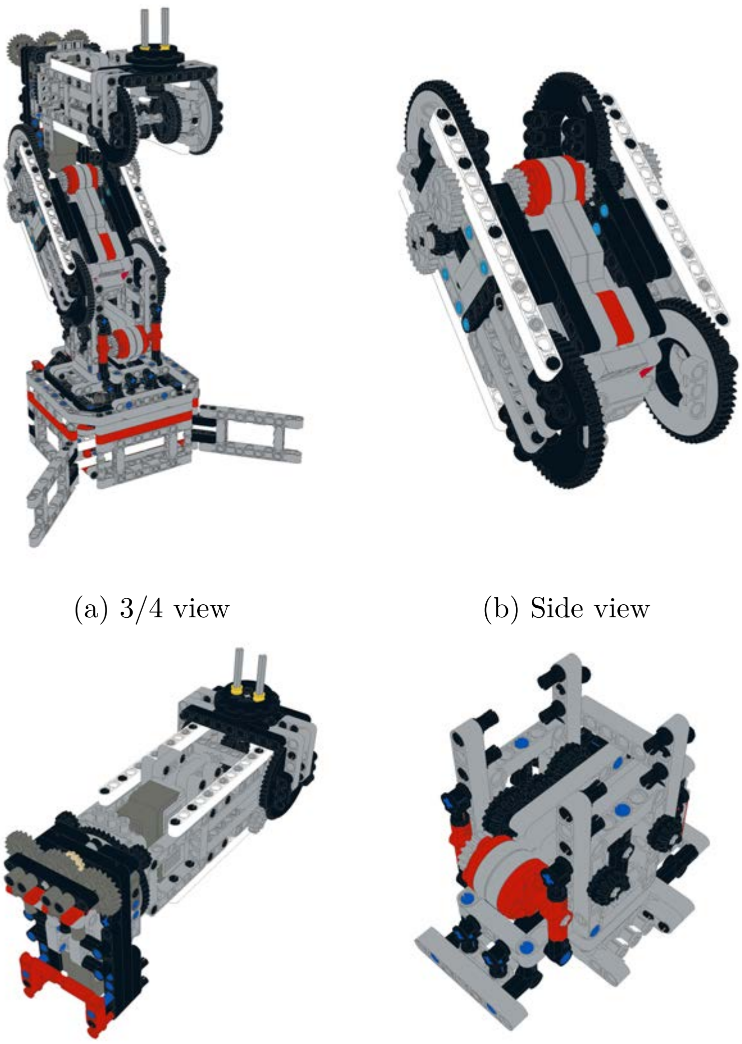


Figure 5.3: Final 6DOF Serial Link Manipulator built with Lego Mindstorms

A major drawback of the servo motors is that they have no sense of absolute angle, meaning that when the system is powered down the motors' internal tachometers retain no memory about their state. An initial attempt to overcome this was to implement a calibration routine upon startup whereby each of the joint servo motors were run to a hard limit and then a process of dead reckoning was employed to calculate and update the state of the arm's joint angles. Indeed, this approach worked relatively well when the arm was fixed to a given frame of reference, namely a table top, and provided the motors did not encounter too much external resistance as they went through their motions. However the tangible hologram system is mobile and the base frame of reference of the system's arms is in a constant state of change, requiring that absolute measurements be gathered for this frame. This, coupled with the fact that the system will encounter a certain degree of external resistance when a user interacts with it and, resultantly, an inevitable amount of positional drift will occur over time and use, rendered the aforementioned solution infeasible. As a consequence the use of Inertial Measurement Units (IMUs) to capture the orientation of the arms' base and limbs was necessary so that the absolute angles of the joints at any moment in time could be calculated. The relative angles of the joints, important for inverse kinematics and for the enforcement of joint limits, are then derived from the absolute positions of the limbs in 3D space. The use of IMUs will be discussed in detail later in this section.

In addition the drawbacks associated with the motors, another issue with using the plastic gears of the Lego Mindstorms system is relatively pronounced gear backlash, as there is a certain amount of play between the gears when they are meshed. Gear backlash is cumulative in gear trains and is noticeable during a change in gear train direction. Some of the gear trains developed for the system's arms contain quite a number of gears and so backlash is noticeable, but acceptable. In order to mitigate the effects of gear backlash a compensation technique was implemented and runs on the EV3 brick. One final drawback of the Mindstorms system is related to one of its key strengths, namely that the unified interface can, at times, make construction a bit constrained and finding the optimal configuration of components and parts can be time consuming.

The first phase in developing the TangHo prototype saw the construction, through the Lego Mindstorms system, of a 6DOF (Six Degrees of Freedom) serial link manipulator with a spherical wrist configuration, as pictured in Figure 5.3. These type of articulated robots are common in industry and the six degrees of freedom, three translational and three rotational, allow for the movement of the manipulator's end-effector into any arbitrary reachable position and orientation. All six joints, namely the waist, shoulder, elbow and the three wrist joints are revolute. The limbs of the arm are modular in nature and the forearm, upper arm and base can all be easily disconnected from each other, creating a certain degree of maintainability when making changes.



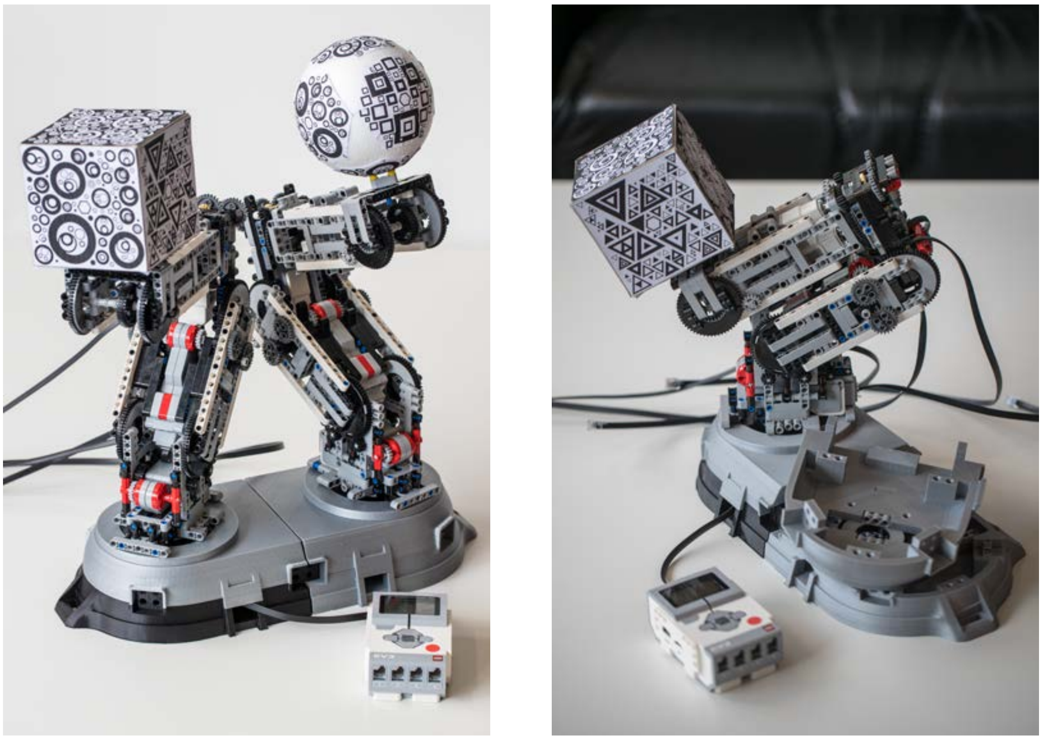
(a) 3/4 view (b) Side view

(c) Close-up of forearm (d) Close-up of shoulder

Figure 5.4: Digital version of arm prototype

A digital version of the final Mindstorms model, elements of which are shown in Figure 5.4, was authored using the LeoCAD[8] software application, which also allows one to create multi-step building instructions from a given model. This was done so that others may build a copy for themselves, thereby building upon the experience gained. The production of this digital model serves as a contribution of this thesis. Figure 5.5 shows the two robotic arms connected to a temporary base platform.

[8] `http://www.leocad.org`



(a) (b)

Figure 5.5: TangHo system

An added advantage of using the Lego Mindstorms system is that the gears, axles, motors and sensors used in the first prototyping phase could be effectively housed in customised 3D-printed casing, thereby making the articulated arms more rigid and lightweight. The various elements of the casing are fastened together using the same Lego pins used in the earlier iteration and components such as servo motors and sensors can also be connected to the housing in a similar manner. The design and production of this customised housing is discussed later in this section.

5.1.2 Inertial Measurement Units

As discussed in chapter four, the mobile aspect of the TangHo system ruled out the use of a more straight forward dead reckoning process to determine the joint angles of the robotic arms. Instead the use of inertial measurement units (IMUs) was required to calculate the absolute orientations of the system's base and arm limbs. The relative joint angles could then be derived from these measurements. The system uses third-party IMU sensors, AbsoluteIMU-ACGs[9] from Mindsensors, pictured in Figure 5.6, which are designed to work with the Lego Mindstorms system. The leJOS api provides a class for reading sensor values from a Mindsensors AbsoluteIMU and this `MindsensorsAbsoluteIMU` class was used in this capacity. In all five such units were used, one housed in the base and two mounted to each arm. Each MindsensorsIMU comprises a triaxial accelerometer, gyroscope and magnetometer and is therefore considered to be a nine degrees of freedom (9DOF) IMU. The basics of each IMU component will first be discussed to provide some context for what will be discussed later.

Figure 5.6: Mindsensors IMU © Mindsensors

Accelerometers measure, or sense, the acceleration of gravity and such information can be used to determine the orientation of the sensor. The acceleration values read from the AbsoluteIMU I2C registers are expressed in milli-G. Here 1 G is the acceleration due to gravity at the Earth's surface, which is equal to $9.81\,\mathrm{m/s^2}$. Should a given axis point toward the Earth's center, then, the value returned by the sensor should, in theory, be 1000 milli-G. The sensitivity of the sensor, which determines the resolution of readings and the maximum range of returned values, can be set to a particular setting, with the accelerometer's sensitivity in this case being set to 2G. As with all the IMU components the raw accelerometer values need to be corrected for any offsets that may be present in the values reported by the component. Figure 5.7 shows the offsets present in values recorded from the accelerometer's three axes while the device remained stationary on a flat surface, with the z-axis pointing towards the Earth's center. A value of equating to 1000 milli-G was subtracted from z-axis readings, such that the readings for all three axes should, ideally, be equal to zero, thereby showing no offsets are present.

[9] `http://www.mindsensors.com/ev3-and-nxt/15-gyro-multisensitivity-accelerometer-and-compass-for-nxt-or-ev3`

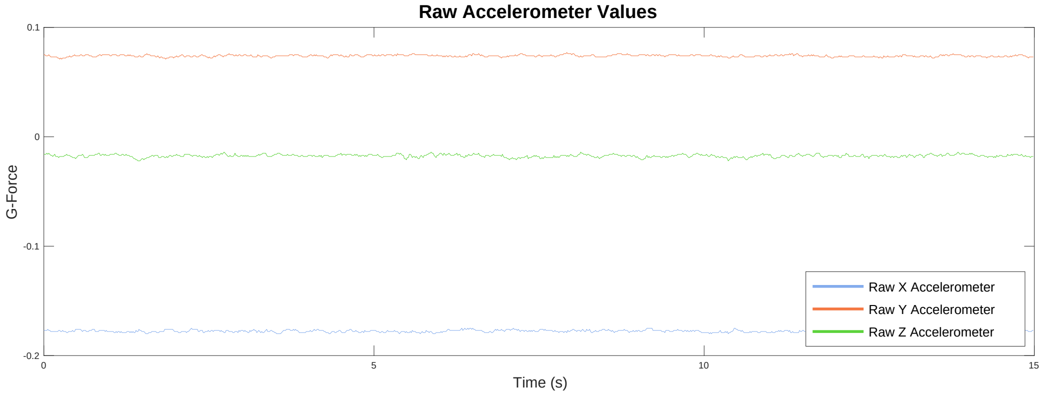


Figure 5.7: Raw accelerometer values

The accelerometer readings in Figure 5.7 were taken over a 15 second interval at a sampling frequency of 60 Hz. As is evident from the figure, offsets are present in the readings of each axis, namely ≈-178 milli-G on the x axis, ≈74 milli-G on the y axis and ≈-16 milli-G on the z axis. Such offsets should be subtracted from any accelerometer readings to help ensure a degree of accuracy. For determining the accelerometer offsets, a sample of 2000 readings were taken while the accelerometer was stationary and then the mean value of these readings was calculated to give the offset value. Figure 5.8 shows the corrected accelerometer values after the offsets have been subtracted. As can be seen in the figure, the readings from the accelerometer, while stationary, now centre around zero.

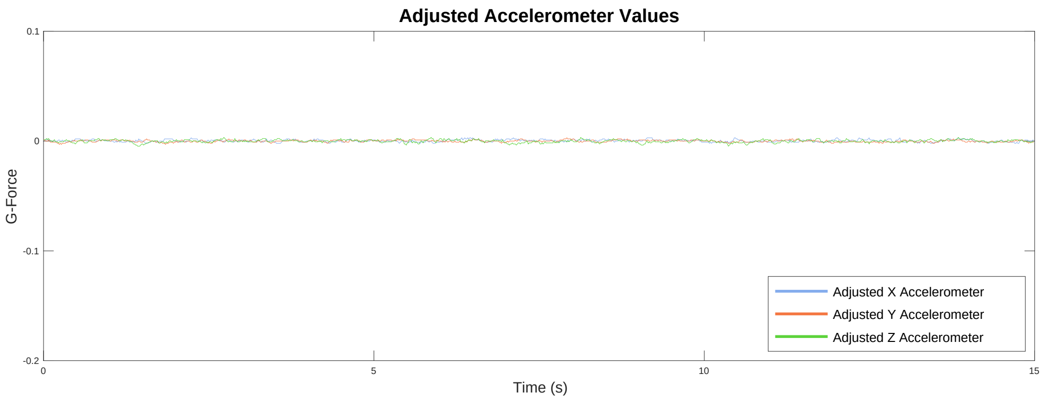


Figure 5.8: Adjusted accelerometer values

A similar treatment of the offsets is required for the raw gyroscope values. Triaxial gyroscopes, such as the one present in the Mindsensors AbsoluteIMU, measure the angular velocity, or the rate of change of angular position per unit time, about three orthogonal axes. If you know the initial angles of the sensor's axes, you may integrate gyroscope values over time to determine the sensor's orientation [37]. One problem with this approach is that measurement errors are also integrated resulting in a drift in the calculated sensor orientation over time. Thus, as noted by Madgwick [37], *"gyroscopes alone cannot provide an absolute measurement of orientation"*.

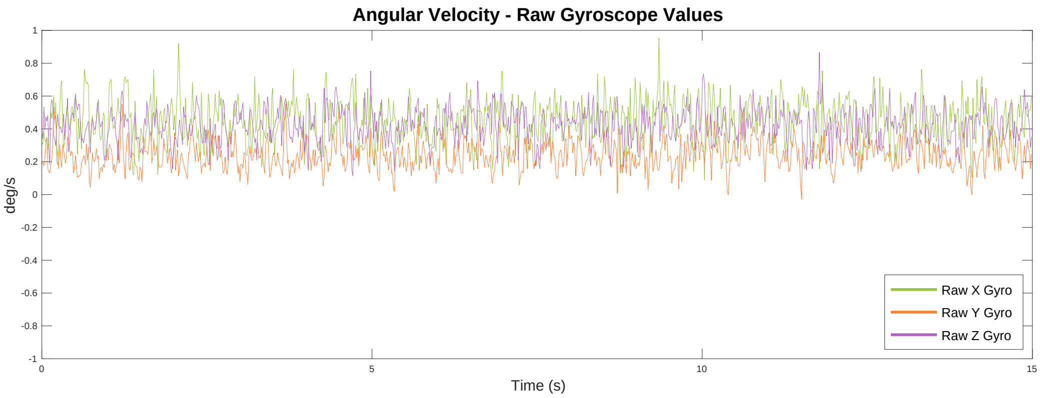


Figure 5.9: Raw gyroscope values

As with the accelerometer, the Mindsensors AbsoluteIMU allows for the setting of different gyroscope sensitivities. For the tangible hologram system the sensitivity was set to 250 degrees/second with the values being read in units of 8.75 milli-degrees/second. Again, there are offsets present in the gyroscope readings, as evidenced in Figure 5.9.

These readings were taken over a 15 second interval at a sampling frequency of 60Hz while the sensor was stationary. Again, ideally, the values for the three axes should all be zero, as the sensor is stationary. The offsets here are ≈451 milli-degrees/second for the x axis, ≈215 milli-degrees/second for the y axis and ≈415 milli-degrees/second for the z axis.

The corrected gyroscope readings are displayed in Figure 5.10. The offsets to be subtracted from the raw gyroscope values were determined by taking the mean of a sample of 2000 readings. As is clear from the figure, the properly adjusted readings now show values centred around zero when the device is stationary.

Correcting the distortion in readings from the triaxial magentometer is a more involved process than that required for the gyroscope and accelerometer. The magnetic field a magnetometer measures comprises both the earth's magnetic field and magnetic fields generated by nearby objects[10]. In taking magentometer readings one needs to account for so-called *hard iron* and *soft*

[10] `https://www.vectornav.com/support/library/magnetometer`

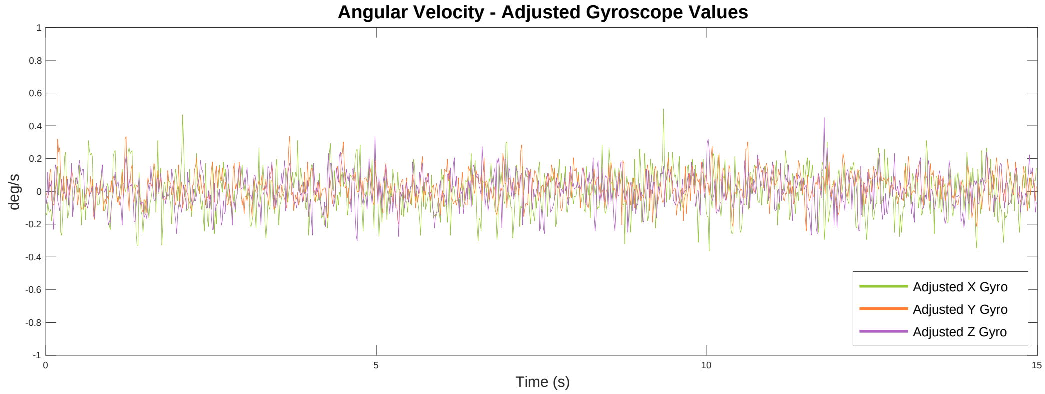


Figure 5.10: Adjusted gyroscope values

iron biases [37, 48], and, given that there will be some additional interference generated by the arms' servo motors, correcting magnetometer data is an important requirement. The source of hard iron biases are magnetic materials in the frame of the sensor itself[10], while soft iron biases are *"sources of interference in the earth frame, [which] cause errors in the measured direction of the earth's magentic field"* [37]. A straightforward routine for correcting hard and soft iron biases is presented by Kris Winer [11], and this solution was adapted for use with the AbsoluteIMU magnetometers.

Figure 5.11 shows raw magnetometer values taken from an AbsoluteIMU sensor at a sampling frequency of 30Hz. As noted by Kris Winer[11], the ideal response surface of a triaxial magnetometer is a sphere centred at the origin. The magnetometer readings plotted here were sampled while slowly rotating the sensor about all three axes and then rotating for a time about arbitrary axes until 2000 samples were collected. It is evident from the figure that there is a misalignment of the responses between axes as these are not centred at the origin. For correcting hard iron biases, Winer proposes to record a sufficient number of magnetometer readings, while moving the sensor slowly in a figure of eight pattern, and record the minimum and maximum values for each axis. The average of these minimum/maximum values can then be subtracted from raw magnetometer readings to center the response surfaces in question at the origin. The offsets for this particular IMU's magnetometer were $\approx$-22.7 milli-Gauss for the x axis, $\approx$6.8 milli-Gauss for the y axis and $\approx$-95.9 milli-Gauss for the z axis.

For correcting soft iron biases, which in effect reshapes the response surfaces such that it more resembles a sphere, Winer suggests taking the minimum/-

[11] `https://github.com/kriswiner/MPU6050/wiki/Simple-and-Effective-Magnetometer-Calibration`

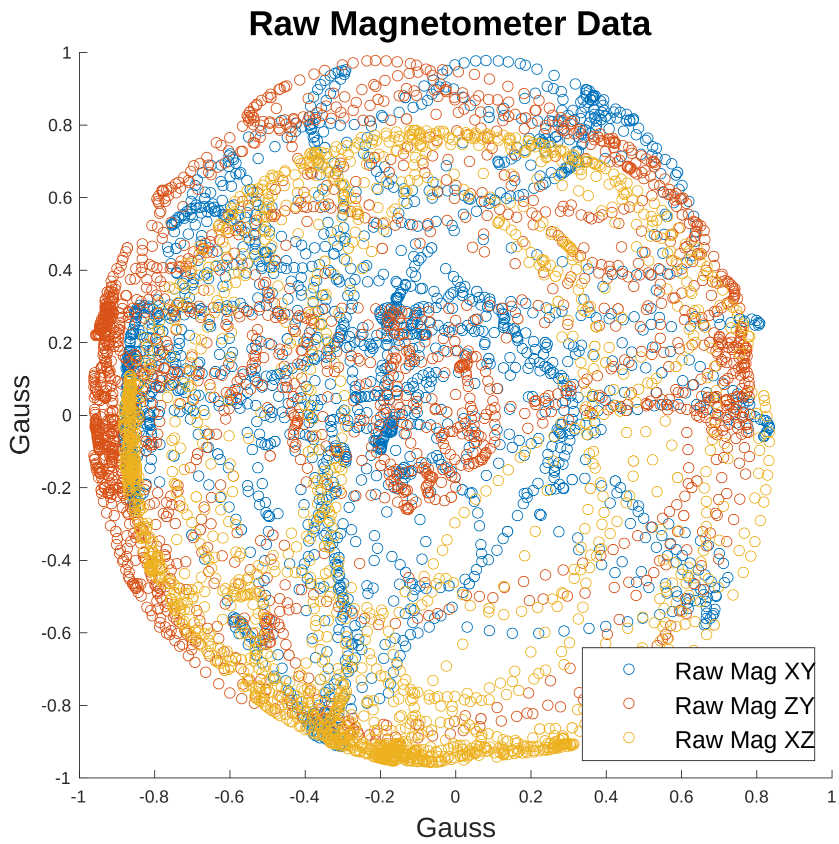


Figure 5.11: Raw magnetometer values

maximum values already recorded and using them to rescale the data in order to *"equalize the response along the three measurement axes"*[11]. These scaling values are calculated by taking the ratio of the average of the minimum/maximum values for each axis and the average of the three axes. Magnetometer readings corrected for hard iron biases are then scaled by by these scale factors to complete the corrections. In this case the scale factors were ≈1.062 for the x axis, ≈0.925 for the y axis and ≈1.022 for the z axis. Figure 5.12 shows the result of the hard and soft bias correction. The response surfaces between axes now align more as each one is centred at the origin and they are slightly more circular in shape.

To derive the orientation of each sensor, then, roll, pitch and yaw angles need to be derived from the accelerometer, gyroscope and magnetometer data, using a sensor fusion algorithm. The accelerometer and magenetometer data are used to correct the drift caused by accumulated error when integrating the gyroscope data. There are a number of different approaches to doing this including the relatively straightforward complementary filter, the more

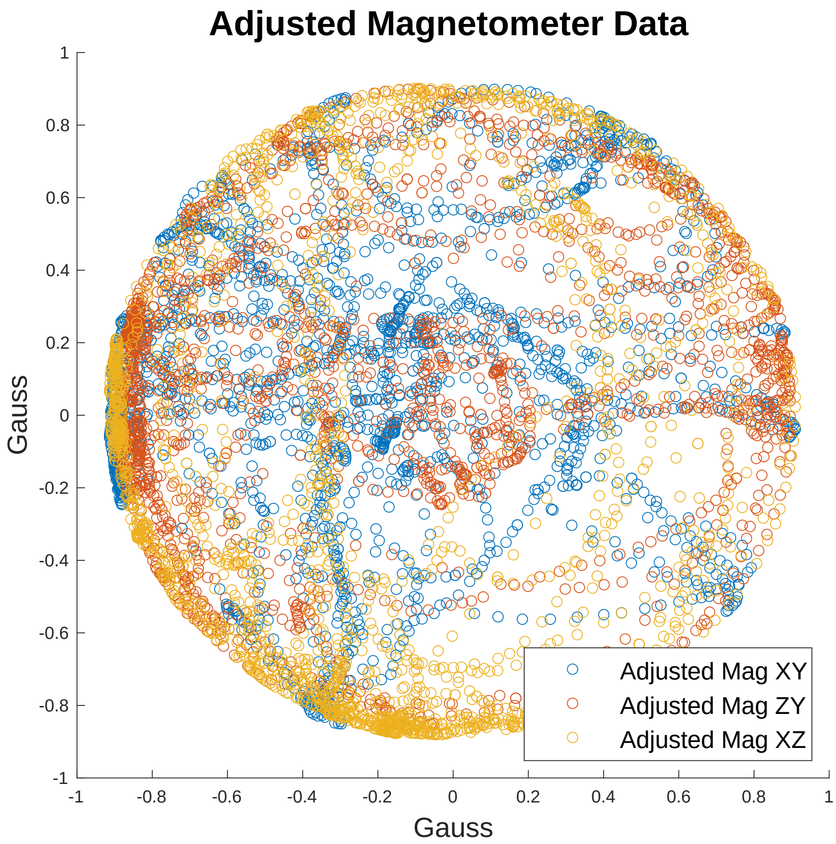


Figure 5.12: Magnetometer values adjusted for hard and soft iron biases

complicated Kalman and Extended Kalman filters and the effecient Madgwick and Mahony filters. The complementary filter was initially investigated as it represented the most straightforward of the approaches. According to Nowicki et al. [36] the principal idea with complementary filters is to apply *"a highpass filter to a biased high-frequency orientation estimate"*, which may be the integration of gyroscopic data mentioned earlier, with a *"low-pass filter to a low-frequency orientation estimate"*, which may be accelerometer or magnetometer data. These two filters work together to provide a stable, less noisy, estimate of a given angle, which is relatively free of the drift associated with using gyroscope data alone.

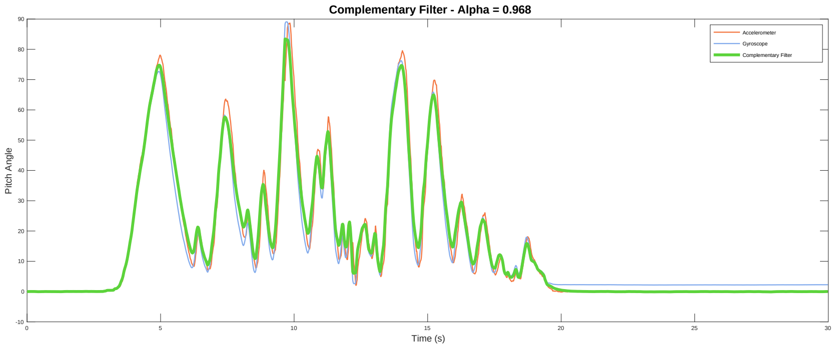


Figure 5.13: Complementary filter

A good overview of the complementary filter is given by Shane Colton [8], who also provides the following straightforward code snippet to illustrate how it may be implemented:

```
angle = (0.98)*(angle + gyro * dt) + (0.02)*(x\_acc);
```

As can be seen from the above snippet, the current angle value is calculated as a combination of a given proportion of the existing angle, with the latest gyroscopic data integrated, and the angle as derived from the current accelerometer data. This accelerometer data may be noisy, thus being unreliable in the short term, so its contribution to a given angle estimation is added in a piecemeal fashion [8]. The proportions α and $(1-\alpha)$, which, in this example, are 0.98 and 0.02 respectively, thus determine how quickly the angle tends towards that derived from the accelerometer data. The alpha parameter can be determined by first choosing a time constant, which determines the relative length of signal the filter will act on, where lower time constants allowing more horizontal acceleration noise to pass through [8]. Once the time constant is chosen, the formula shown in Equation 5.1 is applied to determine the alpha parameter, where τ is the time constant:

$$\alpha = \frac{\tau}{\tau + dt} \tag{5.1}$$

The time constant chosen for the complementary filter plotted in Figure 5.13 was 0.5 seconds, and, given a sampling rate, dt, of $\approx$0.0167 seconds, the alpha parameter was calculated to be $\approx$0.968. The formulas used for calculating the angles from the accelerometer data given in Equations 5.2 and 5.3:

$$\Phi = \tan^{-1}\left(\frac{ay}{\sqrt{ax^2 + az^2}}\right) \tag{5.2}$$

$$\Theta = \tan^{-1}\left(\frac{-ax}{\sqrt{ay^2 + az^2}}\right) \tag{5.3}$$

Figure 5.13 shows the application of the complementary filter on gyroscope and accelerometer readings taken over a 15 second interval at a sampling frequency of 60 Hz. The graph plots pitch angle estimation using accelerometer data, in red, integrated gyroscope data, in blue, and the complementary filter, in green. As can be seen from the graph, the sensor was pitched in number of times in quick succession and then left to remain stationary. Ideally all three graph lines would come to rest at a zero angle reading. However, it is evident that the gyroscopic angle estimation has experienced some significant drift and now contains an error of ≈2.26 °. Figure 5.14 shows this drift in more detail. What can also be seen in this figure is that the angle estimation is not as noisy as that derived directly from the accelerometer data.

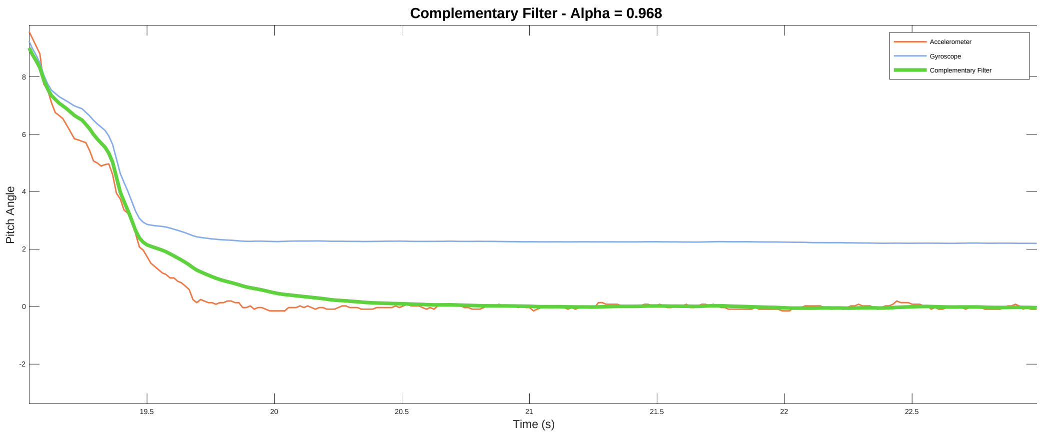


Figure 5.14: Complementary filter close up

Sebastian Madgwick has developed a filter fusion algorithm for the effecient calculation of IMU sensor orientation [37] and has made the source code of his implementation publicly available in MATLAB, C and C#[12]. This algorithm is more accurate than the complementary filter just discussed and represents sensor orientation in quaternion form. A Java implementation of Madgwick's algorithm was sourced[13] and used to determine IMU sensor orientation. According to Madgwick [37], the algorithm is computationally inexpensive and effective at low sampling rates. It takes the adjusted gyro-

[12] http://x-io.co.uk/open-source-imu-and-ahrs-algorithms
[13] https://github.com/pawcio1357/Madgwick-s-filter-fusion-in-Java

scopic, accelerometer and magnetometer data previously discussed and returns a quaternion representation of the sensor orientation when requested. Equations 5.4, 5.5 and 5.6, which appear in Madgwicks paper [37], are used to derive the sensor's yaw, pitch and roll orientation angles, respectively, from the quaternion representation.

$$\psi = Atan2\left(2q_2q_3 - 2q_1q_4\,, 2q_1^2 + 2q_2^2 - 1\right) \tag{5.4}$$

$$\theta = -\sin^{-1}\left(2q_2q_4 + 2q_1q_3\right) \tag{5.5}$$

$$\phi = Atan2\left(2q_3q_4 - 2q_1q_2\,, 2q_1^2 + 2q_4^4 - 1\right) \tag{5.6}$$

5.1.3 3D Printed System

As mentioned previously, one of the advantages of using the Lego Mindstorms system for prototyping was that the non-structural Lego components of the arms, such as axles, gears and servo motors, could be housed in custom 3D printed casing. This casing needed to meet a number of requirements, some rigid and some flexible, and these informed its design. Of paramount importance was the need for the casing to integrate seamlessly with existing Lego Mindstorms components. A modular design facilitating easy assembly/disassembly was also desirable, thereby allowing for maintenance and reconfiguration of gear trains and other Lego components. Ensuring that the casing was as lightweight and structurally robust as possible also ranked high on the list of design considerations.

Casing Design

Designing the casing followed a typical product design workflow, whereby initial sketches were translated into actual scale models using *plasteline* and then to digital form using 3D authoring software. Working with plasteline, a type of modelling clay that maintains its plasticity even after long term exposure to air, proved beneficial in that it allowed for a testing and refining of the sketched designs. In particular, the models produced provided a means by which to gauge how the Mindstorms components, namely servos, gears, sensors and cables, would sit within the housing and, importantly, how the casing components themselves would fit together. However, the benefits of working with plasteline came at the expense of substantial time costs and, due to time constraints, only the base sections were modelled in this way. Another consideration that required attention when designing the casing components was the nature of 3D printing and the restrictions it places on a component design. For example, the build volume of the 3D printer places size limits on individual components and support structures are required for any overhanging parts of the design. With this in mind, then, an effort was made to design the casing components in such a way so as to reduce the amount of overhanging parts, as it may prove difficult, for example, to remove structures supporting hard to access forms such as internal cavities.

(a) Base component bottom section (b) Base component top section

Figure 5.15: Casing Models in Plasteline

Once the casing components were sufficiently refined they were translated into digital format using Autodesk Maya 2017[14], which proved suitable to the requirements of the task. Ensuring that the casing elements could successfully support integration with the Lego Mindstorms elements demanded that the digital models were precise to within less than a tenth of a millimetre. For example, the axle bearing holes, which are essentially cylinders with a stepped opening at either end, are 7.76 mm in length and have a diameter of 4.8mm. The stepped openings have a diameter of 6.2mm and recede a length of 0.8mm back from the face of the opening. This construction allows one to use the holes for fastening pins, if desired, whereby the pins can snap in place and help to hold a construction together. It is evident, then, given the precise nature of these dimensions, that any deviation from them means that pins and axles will either not fit into the holes or be too loose to operate effectively. In addition, to ensure that the casing components themselves

[14]https://www.autodesk.com/products/maya/overview

interlock effectively, one needs to allow 0.1 mm around all interlocking parts. This level of attention to detail saw a considerable amount of time being spent on 3D authoring.

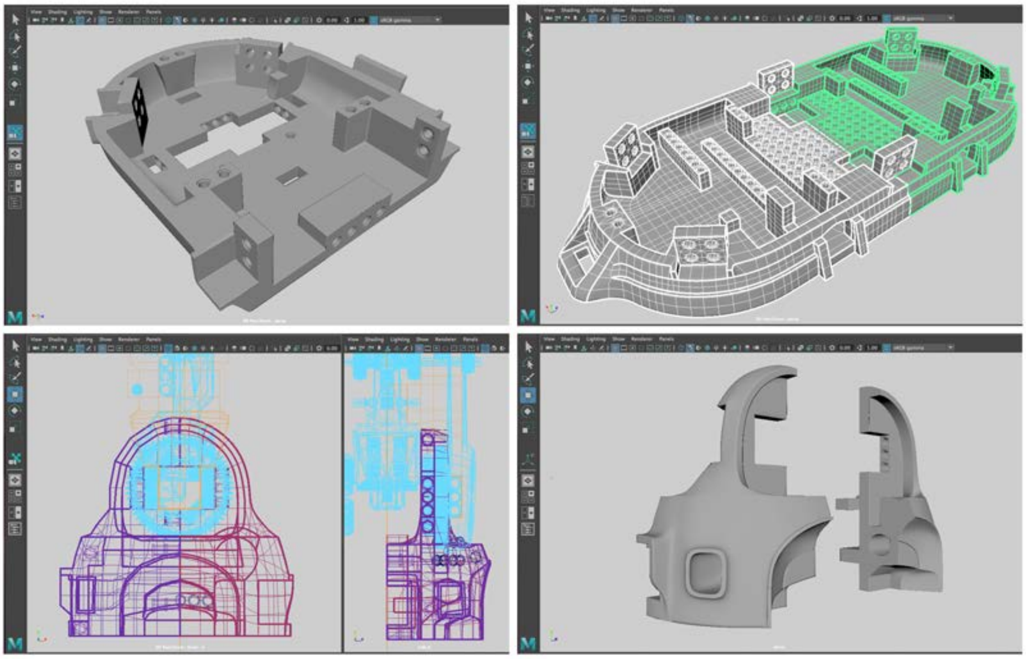


Figure 5.16: Screenshots of 3D Casing Models

3D Printing

The final casing designs were fabricated with an Ultimaker 2+ 3D printer in PLA plastic. The Ultimaker 2+, being a a high-end 3D printer with a layer resolution of up to 20 micron, was highly suited to meeting the high resolution precision required by the casing components. The build volume of the printer was generous, at 223 x 223 x 205 mm, and allowed the base of the tangible hologram system to be comprised of just four large casing components. A typical high quality print may take several hours to a couple of days, depending on the size, and so draft components were printed with low quality settings, reducing the printing time significantly.

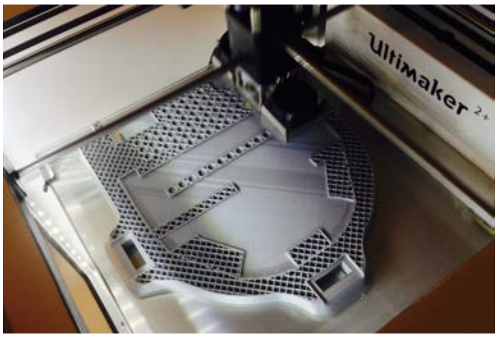


Figure 5.17: Printing a base section

Figure 5.18: Base section close up

Ultimately, due to the time costs associated with plasteline modelling, 3D digital authoring and casing fabrication with the 3D printer, there was not sufficient time to see all the casing components realised. The paper designs of the components, however, may be modelled, refined, authored and fabricated by those wishing to work further on the tangible hologram system.

Figure 5.19: A completed print

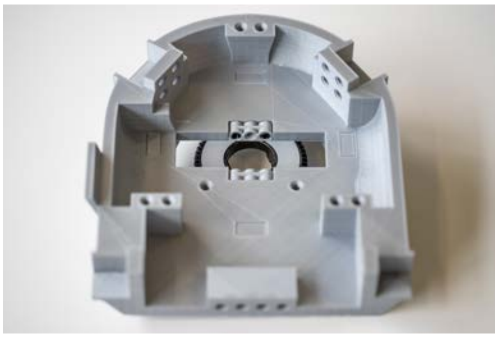


Figure 5.20: Base top section

5.2 Software

The software components of the tangible hologram system are distributed across a number of different system hardware components, with each component bringing its own set of requirements to bear on the software. The system's software components are responsible a varied range of activities such as sensing the absolute limb positions of the system's arms and actuating their servo motors, communication and networking between the different hardware components, performing operations such as forward and inverse kinematics, aligning the coordinates systems of physical space and HoloLens space and providing interaction with digital content.

5.2.1 System Dataflow and Software Architecture

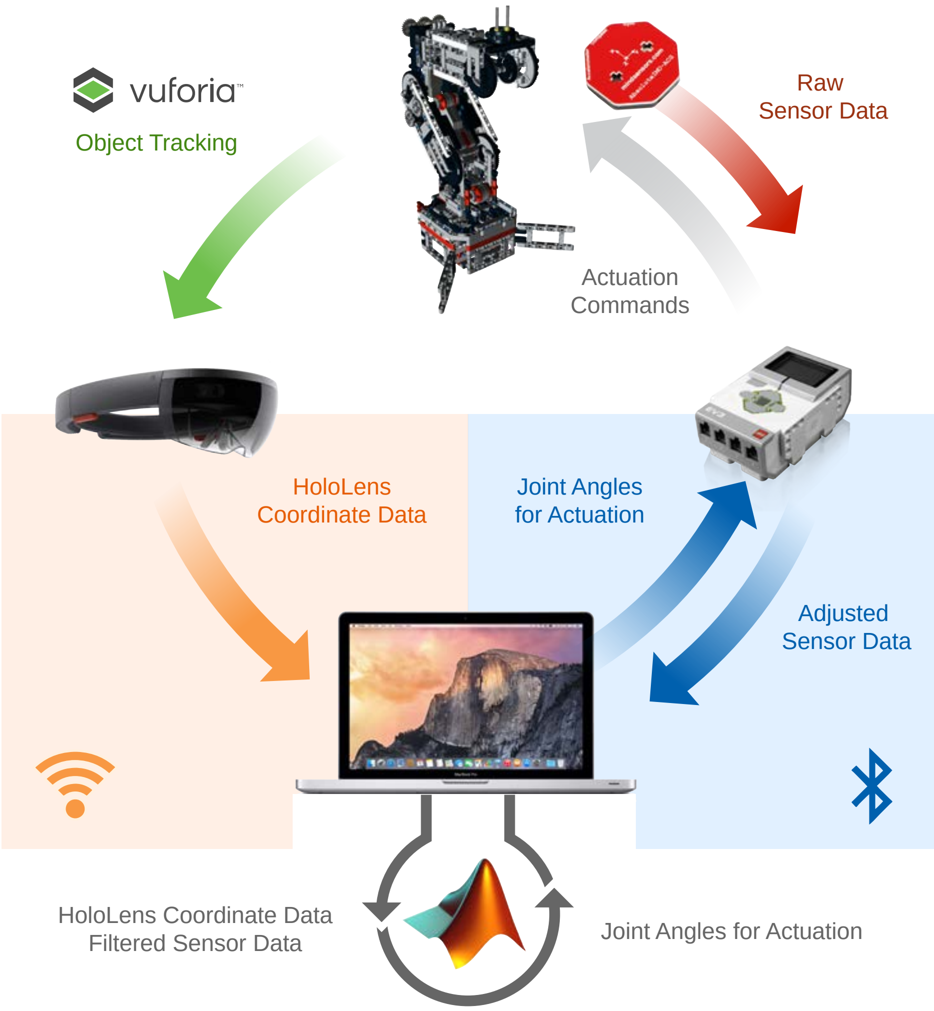


Figure 5.21: Tangible holograms dataflow

At this point it may prove useful to discuss how data flows through the tangible hologram system, as this can serve to provide a useful overview of the system as a whole and give some context to the software components detailed later. Figure 5.21 shows where the various data are generated and how they flow through the system. Applications running on both the HoloLens and the EV3 bricks communicate with the main application through its associated server, both of which run on the MacBook Pro. The main application can be seen, then, as acting as a mediator between the HoloLens and the system's arms. Coordinate data generated by the HoloLens, such as the position of arms' end-effectors in HoloLens space and coordinates relating to which digital object a user is interacting with, are passed to the main application so that it can be used in kinematic calculations performed by MATLAB. In tandem with this stream of information, the main application receives adjusted sensor data from EV3 bricks, which they, in turn, read from the registers of the IMU sensors located on the system's arms. This sensor data are also required for kinematic calculations, as the arms' joint angles can be calculated from this information. Once the joint angles for actuation are calculated, using inverse kinematics, they are sent back to the EV3 bricks, which are responsible for running the actuation routines. Updated end-effector positions are then visually observed by the HoloLens, though the utilisation of Vuforia library components running on the HoloLens application, and limb orientation is sensed through the IMU sensors on the robot arms, thereby generating part of the data for the next cycle through the system. For TangHo system class diagrams please see the Appendix.

5.2.2 Communication

Networking plays an important part in the tangible hologram system, as the HoloLens and EV3 components constantly communicate with the main application's server running on the MacBook Pro. All system communication is built using connection-oriented TCP sockets, thereby providing reliable bidirectional links between communicating system components. The MacBook Pro communicates with the HoloLens via WiFi and with the EV3 bricks via Bluetooth. A network interface with a manually assigned static IP address was set up on the MacBook Pro, through which the HoloLens and EV3 clients connect to the main application server.

A number of classes were implemented to help facilitate the setup of socket servers and clients and most of these are Java-based, with the exception of the `TCPClient` class which was implemented in C# as it runs on the HoloLens. The `SocketServer` class handles the process of setting up a server socket and accepting connections from clients, in this case the HoloLens and EV3 applications. Buffers, in the form of `ArrayBlockingQueues`, hold incoming and outgoing messages until the main application or the server's clients are ready to process them, and are needed as this server is responsible for handling a number of different streams. The `SocketServer` also maintains a registry of its active clients, in this case connections from the HoloLens application and from the two EV3 applications responsible for the systems two arms.

Clients of the `SocketServer` are represented by instances of the `Client` class. Upon accepting a connection from a client socket the `SocketServer` will instantiate a `Client` object, passing in the accepted socket connection and a reference to itself. This instance is then stored in a registry of clients, as previously mentioned, with the client's connection address as the key. This key is then used whenever an outgoing message needs to be routed by the server to the appropriate client instance, which then sends the message on its own socket's output stream. The `Client` class contains a nested `Receiver` helper class, which runs on its own thread and blocks until there is data on the socket's input stream. Once a message is received, it's placed in the `SocketServer`'s in buffer to be handled by the server instance.

The `SocketClient` class is used by applications wishing to connect to the aforementioned server and is used, therefore, by the applications running on the EV3 bricks. It is responsible for making the connection to the server and for sending and receiving messages from it. Similar to the `SocketServer` class, it contains two buffers for incoming and outgoing messages, again implemented as `ArrayBlockingQueues` and client code can read from or place data in these buffers. `SocketClient` contains two nested helper classes implementing the runnable interface, namely `Sender` and `Receiver`, both running on separate threads. The `TCPClient` class utilized by the HoloLens application operates in a similar manner to the `SocketClient` just discussed.

To ensure consistency and efficiency across communications, JSON was selected as the transport mechanism for all messages cycling through the system. It is compact, lightweight and, owing to the fact that it is language independent, allows for straightforward interoperability across platforms. The system's Java-based code uses the `java-json` package while, the `C#` code running on the HoloLens uses `SimpleJSON`. As another measure to ensure communication consistency, a list of application constants, stored in the `AppConsts` class is utilised by all system components to correctly tag and identify messages. All JSON messages have a `type` field, which is assigned one of these `AppConsts` values as an identifier.

5.2.3 Main Application

As previously mentioned, the system's main application runs on the MacBook Pro and is written in Java. The entry point for the application is the `MainController` class which instantiates the components necessary to interact with the rest of the system. The `MainController` instantiates an instance of the custom `SocketServer` class, which runs within its own thread and, as previously mentioned, handles the process of setting up a server and sending messages to and receiving messages from clients, in this case the HoloLens and EV3 applications. It also instantiates a number of `ArrayBlockingQueues`, which act as data buffers for information generated by the HoloLens and the two EV3 applications responsible for each of the system's arms.

The MainController runs an event loop within a separate thread to handle incoming messages from the HoloLens and EV3s, passing on messages to those worker threads responsible for further processing. The forward and reverse kinematics calculations are performed in MATLAB and Java programs, by setting up a session through the MATLAB Engine API for Java, can use it as a computational engine. The MatlabRobot class wraps an instance of the MATLAB engine and handles engine initialisation and creating an instance of the SixAxisRobot class required for performing calculations on the MATLAB side. The MatlabRobot class handles all communication with the engine, preparing and sending the execution statements and passing the results on to its MainController client.

The MainController uses instances of the runnable SensorProcessor class to process IMU sensor information coming from the EV3s and instantiates one SensorProcessor instance per arm, assigning each one to its own thread. Processing sensor information on the MacBook Pro frees up the resources of the EV3 bricks for activities such as reading and sending sensor data and receiving and executing servo motor actuation commands. The SensorProcessor class takes an ArrayBlockingQueue, for holding sensor readings from a given arm, as a construction argument and is responsible for processing data from this queue for the lifetime of the application. In applying a filter to the sensor data, the SensorProcessor can use any class implementing the IMUFilter interface and, at present, there are two classes, ComplementaryFilter and MadgwickFilter, which were discussed in an earlier section, implementing this interface. IMUFilter requires the implementation of the overloaded update method, which takes gyroscope, accelerometer and, optionally, magnetometer data, while the setSamplePeriod method allows one to change the sampling period used by the filters on the fly. The IMUFilter interface also requires the implementation of the getEulerAngles method returning euler angles calculated from the filtered sensor data.

5.2.4 EV3 Application

The EV3Controller class is the entry point for the application running on the EV3 bricks and is responsible for executing all tasks relating to monitoring and controlling the system's arms. For communication with the main application server, the EV3Controller uses an instance of the of the SocketClient class, which sets up the server connection. New messages received from the server are stored in the SocketClient's in buffer, which, in turn, are read by an instance of the EV3Controller's nested Reader class, which is a small helper class running within a separate thread.

One of the main responsibilities of the EV3 application is to read orientation data coming from the IMU sensors mounted on the base and limbs of the robotic arms and send on a corrected version of that data to the main application for further processing. Each arm has two mounted IMU sensors, one on the upper arm and one on the forearm, with the base information coming from a shared IMU sensor. The `SensorHandler` class is responsible for coordinating the retrieval of sensor information for all of the IMU sensors mounted on the arms and base of the haptic unit. It implements the `Runnable` interface and the EV3 controller creates an instance of this class and runs it on its own separate thread. To get the readings from each individual sensor, the `SensorHandler` class is passed an array of `SensorReader` instances. Each `SensorReader` class takes care of the reading of values from a given sensor's registers and applying corrections to compensate for sensor offset and bias. As each sensor has different offsets and biases, a `SensorReader` instance is required for each one. Pre-computed offsets and biases for each sensor are stored in the static `sensorOffsetsMap` hashmap so that these values do not have to be computed every time the system starts up. The `SensorReader.read()` method is called periodically by the `SensorHandler` class to get the latest values. Initially it was possible to modify the leJOS `MindsensorsAbsoluteIMU` class so that it used I2C high-speed mode, resulting in accurate $\approx$2 ms reads from the sensors registers. However, when the sensor's cable was extended using an extender, which was also sourced from Mindsensors, we noted discrepancies in the readings and had to revert back to the class's original settings. This resulted in much slower reads of $\approx$12 ms.

The `SensorReader` class also has a `recalibrate()` method should one wish to generate fresh values for use in sensor offset and bias compensation. The method will calculate offset values for the accelerometer, gyroscope and magnetometer as well as magnetometer scaling values. As the sensors should use the same sampling frequency, this value is stored in the static `samplingHz` value, which is accessible through a getter and setter. Once the `SensorHandler` instance has completed a successful round of `SensorReader.read()` calls, it sends the data to the `MainController` application running on the Macbook Pro, through its `SocketClient` instance.

In terms of arm actuation, then, server messages containing joint angles for actuating the arms into the required poses are passed to appropriate instances of the `ArmController` class by the `EV3Controller`. The `EV3Controller` instantiates two `ArmController` instances, one for each arm, and delegates the responsibility for arm actuation. The `ArmController` class uses many of the classes provided by the leJOS API to carry out its functions.

Before discussing how these classes are utilised, it is worth mentioning how leJOS facilitates inter-brick communication. This is important as each arm requires six motors to actuate all of its joints and, as a single brick contains only four motor ports, two EV3 bricks are required to work in concert to achieve the actuation of one arm. leJOS enables inter-brick communication by allowing one to set up a Personal Area Network (PAN), which, if using Bluetooth, can have up to eight EV3 brick members. One brick is designated as an EV3 access point and essentially acts as a controller for the PAN, handling tasks such as the assignment of IP addresses, for example. Once bricks are connected to the network they can communicate with each other through TCP [15]. The PAN is configured on the EV3 brick designated to be the access point through the GUI provided by the leJOS operating system. Fixed IP addresses can be assigned to each of the EV3 bricks on the network and the access point for the PAN in the tangible hologram system is assigned the IP address `10.0.0.1`. Bricks wishing to connect to this network are then configured to be Bluetooth clients, again through the leJOS GUI, and, in order to connect to an EV3 access point via Bluetooth, the brick must first be paired with the access point EV3 [15].

Once the PAN is set up, any EV3 on the network can be accessed by a given program, allowing a program executing on one brick to harness the resources of another, in this case the two extra motor ports needed for executing the actuation of one arm with one `ArmController` instance. The `ArmController` class is passed the brick names and port numbers of those motor ports responsible for the actuation of one of the system's arms as constructor arguments. The `ArmController` then makes remote requests to each of these bricks through instances of the leJOS `RemoteRequestEV3` class, each of which it instantiates by passing the IP address of a given brick as a constructor argument. Brick IP addresses can be retrieved by first calling the static leJOS `BrickFinder.find` method and passing in the name of the brick as an argument. This method call returns a table of the addresses that can be used to contact that particular brick and the first IP address in this table is supplied to the `RemoteRequestEV3` class as a constructor argument. The resultant `RemoteRequestEV3` object, which is then assigned to a private instance variable, can be used to instantiate the leJOS `RegulatedMotor` instances needed to perform tasks such as setting the servo motor's speed and executing a given motor's rotation commands. The `ArmController.move` method is called by the `EV3Controller` when it wishes to execute an actuation. It takes the joint angles that each of the joints need to rotate to and contains a rudimentary gear backlash compensation mechanism.

[15]`https://lejosnews.wordpress.com/2015/02/11/`

5.2.5 Forward and Inverse Kinematics

The fast, efficient and correct calculation of forward and inverse kinematic solutions is an essential requirment of the tangible hologram system and, given the time constraints associated with the project, it was decided to work with a well-established library that could provide the functionality required. Three resources with kinematic solvers were explored to see if they would be a suitable option for the system, namely Peter Corke's Robotics Toolbox for MATLAB [9], CALIKO [26], a Java-based library implementing the FABRIK algorithm [2] and MATLAB's own Robotics System Toolbox, which contains a wide range of functionality for developing autonomous robotic applications for various types of robots, including manipulators [16].

Ultimately, Peter Corke's Robotics Toolbox was chosen as it was the best fit for the project at hand. The toolbox was first introduced in 1995 and has seen numerous realises since then, with the most version, 10, being realised in 2017. With over twenty years of development, then, it can be said that the code is quite mature and robust [17]. The library's author, Peter Corke, is a professor of robotic vision at the Queensland University of Technology and the toolbox is used in teaching robotics at this and other universities around the world. One of the benefits of the toolbox is that the source code is accessible, well commented and written in a pedagogical spirit, so one can easily follow the routines and have a clear understanding of how they work. Corke, acknowledges that this clarity may come at the expense of some efficiency, but notes that routines may always be re-written, tweaked or otherwise optimised by those using them [17]. In any case, the toolbox routines used by the system proved fast enough for the purposes of testing the system prototype so any relative loss in efficiency was not an issue. However, those working on future iterations of the system may want to tweak and optimise some of the routines as suggested. Another advantage of selecting this toolbox over other options is that Corke has published an introductory text to robotics [9], which uses routines from the toolbox to illustrate the concepts and material covered in the book. This is beneficial in getting a fuller understanding of how the toolbox itself works in addition to getting a grounding in those concepts behind the toolbox routines.

The primary class of interest in the robotics toolbox, in terms of the tangible hologram system, is the `SerialLink` class, with which one can build a model of a serial link manipulator robot. Corke [9] gives a good overview of serial link manipulators in his introductory text to robotics, and this proved a very useful reference when modelling the system's robotic arms using the

[16] `https://uk.mathworks.com/products/robotics.html`

[17] `http://petercorke.com/wordpress/toolboxes/robotics-toolbox`

toolbox. A serial link manipulator is made up of a set of rigid bodies, or links, connected by joints, thus forming a chain and each joint in the chain can either be *prismatic* (translational), or *revolute* (rotational). Each joint offers one degree of freedom and, in the case of the six degrees of freedom (6DOF) manipulator modelled for the tangible hologram system, all joints are revolute. The pose of the end-effector, which is the tool end of the robot arm, is then *"a complex function of the state of each joint"* [9].

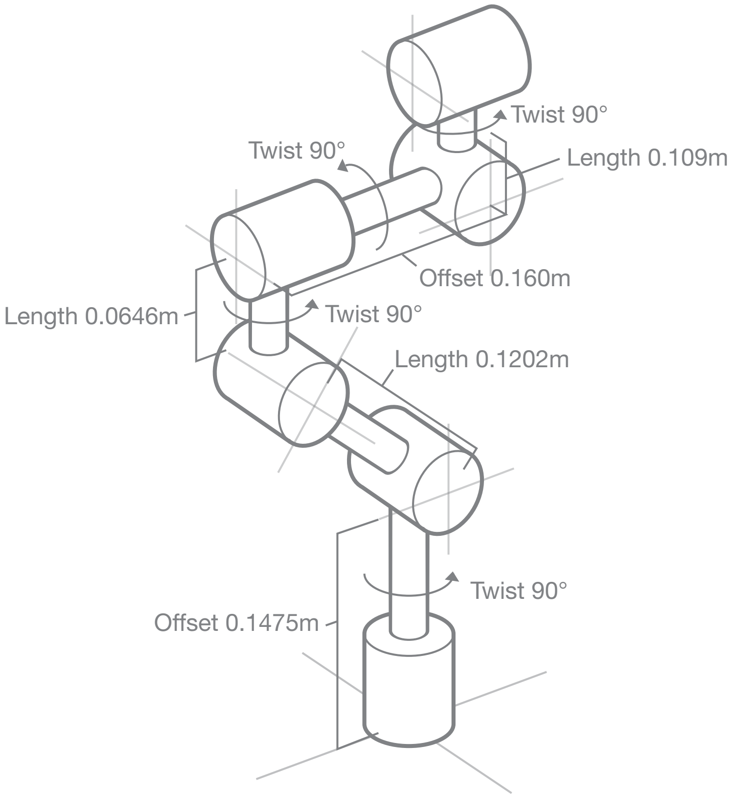


Figure 5.22: Tangible hologram system arm modelled with Denavit-Hartenberg parameters

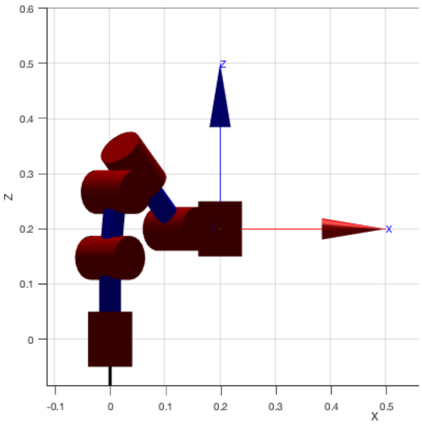
Denavit-Hartenberg parameters, which allow for a *"systematic way of describing the geometry of a serial chain of links and joints "* [9], were used to model the robot arm required by the system. Links can be described by their length and *twist*, while joints can be described by their *link offset*, which is the distance between *"one link coordinate frame to the next along the axis of the joint"*, and their joint angle rotation relative to the next link about the joint axis [9]. Figure 5.22 illustrates the serial link manipulator model for one of the system's arms in its default pose with joint angles, from base to end-effector, of 0°, 135°, -45°, 0°, 0°, 90°. The relevant Denavit-Hartenberg parameters are denoted and where no parameters are given, one can assume that the value is zero. When modelling with the toolbox the measurements for length are unit agnostic, however, since the unit of measurement for HoloLens applications is the metre, all measurements for the serial link manipulator are given relative to a metre.

The `SerialLink` class takes an vector of links as a constructor argument, with each one of these links being an instance of either the `Link`, `Prismatic` or `Revolute` classes. In the case of the manipulator modelled here, an vector of `Revolute` instances was passed in, as all the joints are revolute and it is convenient to adopt this approach. Listing 5.1 shows how the links are instantiated and passed to the `SerialLink` class, where `'d'` is the link offset, `'a'` is the link length, `'alpha'` is the link twist and `'qlim'` are the joint limits.

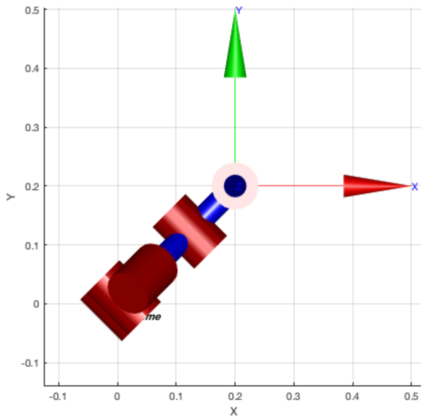
Listing 5.1: Passing an Array of Links to a Serial Link Manipulator

```
% Base Joint
Links(1) = Revolute('d', 0.1475, 'a', 0, 'alpha', 90 * arm.toRad, ...
    'qlim', [-90 90] * arm.toRad);
% Shoulder
Links(2) = Revolute('d', 0, 'a', 0.1202, 'alpha', 0, ...
    'qlim', [45 150] * arm.toRad);
% Elbow
Links(3) = Revolute('d', 0, 'a', 0.0646, 'alpha', 90 * arm.toRad, ...
    'qlim', [-45 90] * arm.toRad);
% Wrist 1
Links(4) = Revolute('d', 0.160, 'a', 0, 'alpha', 90 * arm.toRad, ...
    'qlim', [-90 90] * arm.toRad);
% Wrist 2
Links(5) = Revolute('d', 0, 'a', 0.109, 'alpha', 90 * arm.toRad, ...
    'qlim', [0 180] * arm.toRad);
% Wrist 3
Links(6) = Revolute('d', 0, 'a', 0, 'alpha', 0 * arm.toRad);
arm.sl = SerialLink(Links);
```

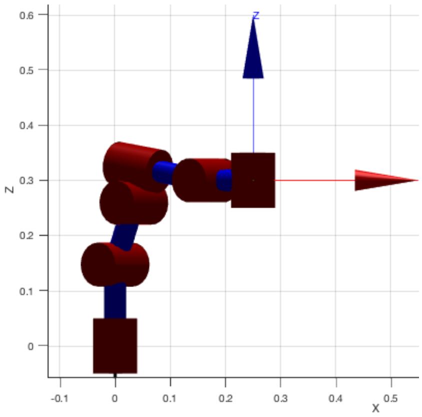
Several kinematic solvers can be called on the `SerialLink` class once it is instantiated and the `SerialLink.fkine` and `SerialLink.ikcon` methods are utilised by the tangible hologram system to perform forward and inverse kinematics calculations respectively. Forward kinematics is the process whereby the pose of the manipulator, and the resultant pose of the end-effector, are calculated from known joint angles and is a relatively straightforward process requiring little computation. The `SerialLink` class exposes one method, `SerialLink.fkine`, for performing forward kinematic calculations. Inverse kinematics, a process whereby the joint angles required to configure the end-effector to a known pose are to be calculated, on the other hand, is a little more complicated. Here the `SerialLink` class exposes many methods, each using a different approach for calculating inverse kinematic solutions. The `SerialLink.ikcon` numerical solver method, which can be called on robots with arbitrary degrees of freedom, was chosen as it gives good results, does not contain any prototype code under development, and, importantly for this system, respects joint limits. Figure 5.23 shows two solved inverse kinematic problems, where the `SerialLink.ikcon` method was supplied with the coordinates 0.2, 0.2, 0.2, in the case of the top two sub-figures, and 0.25, -0.2, 0.3, in the case of the bottom two. As is evident from these figures, this kinematics solver gives accurate results.



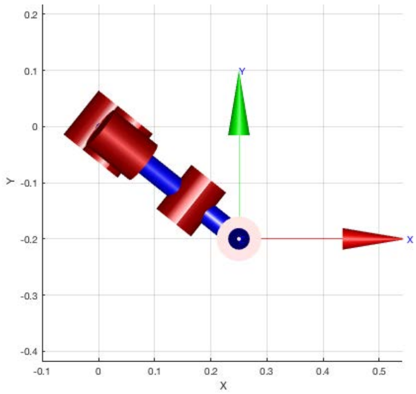
(a) Solving for 0.2, 0.2, 0.2 - Side View



(b) Solving for 0.2, 0.2, 0.2 - Top View



(c) Solving for 0.25, -0.2, 0.3 - Side View



(d) Solving for 0.25, -0.2, 0.3 - Top View

Figure 5.23: Inverse Kinematic Solutions with `SerialLink.ikcon`

A wrapper class for the `SerialLink` manipulator, the `SixAxisRobot` class, was implemented to encapsulate additional data and functionality, with most of the interaction with the robotics toolbox being mediated through this class. This helps to make the utilisation of the toolbox more convenient and the class exposes a simple interface to client code, while hiding unnecessary initialisation and processing steps. If no arguments are supplied to the `SixAxisRobot`'s constructor, the class, upon instantiation, creates the default set of links, as shown in listing 5.1, and passes them to a `SerialLink` constructor, assigning the resultant `SerialLink` instance to a class variable for later use. The default configuration for the robot will also be set to the aforementioned home position angles. Should a different set of links be required, the `SixAxisRobot` constructor can take two vectors, one with the six required links and one with six default configuration angles, and use these when instantiating the `SerialLink` object instead. The

`SixAxisRobot.solveIK` method, which in turn utilises the `ikcon` method, is used to perform the inverse kinematics calculations.The method takes the physical robot's current joint angles, as derived from the IMU sensor data, the position of the robot's physical end-effector position in the Hololens application's coordinate space and the coordinates that the end effector should move to. When calculating the inverse kinematics solution for a given set of coordinates, the `SixAxisRobot.solveIK` method first needs to set the base of the `SerialLink` instance so that it represents the actual cartesian pose of the base in the HoloLens application's space. To do this, a homogeneous transform, which is a 4 x 4 matrix containing rotational and translational information, needs to be constructed and then set as the `SerialLink` instance's base frame. The homogeneous transform contains an orthonormal 3 x 3 matrix representing the rotational component of the pose and this can be calculated by supplying the pitch, roll and yaw angles, derived from the IMU sensor data on the robot's physical base, to the toolbox's `rpy2r` method. The translation component of the homogeneous transform requires the coordinates of the robot's physical base relative to HoloLens application's coordinate space and the `SixAxisRobot.setHololensBaseCoords` method is used to calculate this. First, an end-effector position, which considers a base at the origin, is calculated using forward kinematics and the coordinate component of the returned homogeneous transform is then extracted and used to calculate an offset vector with inverted values. This offset vector is then added to the reported coordinates of the physical end-effector and the resulting vector is combined with the aforementioned rotational matrix using the toolbox's `rt2tr` method, which takes a rotational matrix and a translation vector as arguments. The `SerialLink` instance's base frame is then set with this transform and the inverse kinematics calculations are performed.

5.2.6 Microsoft Hololens Application

HoloLens applications are typically developed with the Unity cross-platform game engine, where 3D objects, which will later become holographic content, can be created and animated. Indeed, Unity is one of the tools Microsoft recommends for developing mixed reality applications and version 5.6.0f3 was used to develop the tangible hologram application for the HoloLens. Scripts for use in Unity applications are written in C# and a comprehensive well-documented API is provided by Unity to facilitate application development. The Unity editor interface presents a rich set of functionalities to the user, with a stage for placing and composing objects and an inspector for accessing and editing object properties.

Each interactible hologram comprises one or many *GameObjects*. The GameObject constitutes one of the most important elements in Unity application development, acting as a container for grouping different components and, indeed, other GameObjects[18]. In this way, some complex hierarchies may be generated. A GameObject can be also be an empty entity, merely acting as a container for related scripts. Such objects usually act as managers for different functionalities, such as spatial mapping management or handling input. The tangible hologram application running on the HoloLens has two such GameObjects, one acting as an input manager, with attached scripts for handling gesture and speech input, and one holding the application's `MainController` script.

Those GameObjects representing holograms contain meshes that form their visual elements and scripts that provide interactivity and behaviour. Holographic GameObjects also contain *Mesh Collider* and *Rigidbody* components to enable collision detection, while their local and global position, orientation and scale can be retrieved from the *Transform* component. As previously mentioned, GameObjects can house complex hierarchies and scripts may be attached at any level, thereby affording reuse, modularity and a rich interplay between scripts. Thus specific GameObject behaviour can be built up using more generic scripts and a nice feature of the Unity editor is that the value of serialised fields on an attached script can be set through the inspector window. This allows one to attach a script to multiple GameObjects and configure each one through the editor as required, without the need to write extra initialisation code.

When developing applications for the HoloLens, the MixedRealityToolkit[19] is a useful addition to any project and the library was utilised in this application. The MixedRealityToolkit is an open source project with contributions

[18] urlhttps://docs.unity3d.com/Manual/GameObjects.html

[19] `https://github.com/Microsoft/MixedRealityToolkit-Unity`

from a large community of developers and comprises a collection of scripts and so-called *prefabs* to help speed up the development process. Prefabs constitute a way to store pre-constructed GameObjects such that several instances may be generated from the same template. In addition they allow for the dynamic instantiation of GameObjects during runtime. The prefabs in the MixedRealityToolkit are convenient to use and the tangible holograms application uses the `HoloLensCamera` prefab as its main camera. This has all the necessary optimal settings for working with the HoloLens headset already preconfigured.

The application developed to demonstrate the tangible hologram system uses two different modes to interact with the haptic unit and these modes can be toggled through voice commands. The first mode detects if tangible holograms are within a given proximity of a user's hands. To achieve this the user's hands are first tracked when they are in the so-called *ready state*, one of a couple of gestures the HoloLens will recognise for hand tracking. Once a given hand is being tracked an invisible sphere with a radius of 50 cm is positioned at the centre of the hand and will move with the hand during tracking. This sphere GameObject has a Sphere Collider and Rigidbody component attached to it such that when it collides with tangible hologram objects, which also have attached Collider and Rigidbody components, a collision event is triggered. When a collision event is fired on a tangible hologram object it is handled by the `TangibleObject` script, which is attached to every holographic GameObject. The collision event handler in the `TangibleObject` script passes the coordinate data of the GameObject to the application's `MainController` instance for further processing. The second interaction mode causes the arms of the haptic unit to directly track the hands and is used to test and evaluate the system. In this mode the hand positions themselves are registered with the `MainController` instance.

The `MainController` class, then, is the principal class of the tangible hologram application, responsible for sending positional information to the main application running on the Macbook Pro. The class instantiates an instance of `TCPClientConnection` to handle making a connection to the server running on the Macbook Pro and to send the aforementioned positional information to this server. The `MainController.Update()` method checks if there is any new data to send, for example an updated end-effector position, and, if so, sends this information in a JSON message. Once a message is sent, a timeout of 25 ms is started, such that no new data is sent to the server until this timeout has expired. This creates less processing overhead for the applications running on the MacBook Pro and EV3 bricks. To receive updates on hand positions, the `MainController` instance, which implements the `IObserver` interface, registers itself as an observer of the custom `HandDetection` class instance, which, in turn, implements the `IObserverable` interface. As the `HandDetection` instance receives updated hand position information during hand tracking, it calls the `OnNotify()` method of its observers, one of whom is the `MainController`, and passes along the updated positions.

Vuforia

The tracking of the arms' end-effector positions is an important requirement for the tangible hologram system. Knowing the position of either of the system's two end-effectors at any given time allows the system to calculate the position of the haptic unit's base in the HoloLens application's coordinate space. This is essential in a mobile system as the position and orientation of the unit's base are in continual flux. Therefore, the tangible hologram application running on the HoloLens requires a means to recognise the end-effectors and track their position. Vuforia[20], a company who specialise in computer vision for Augmented Reality (AR) applications, offers a free Unity package designed to work with the Microsoft HoloLens and this was used to achieve end-effector recognition and tracking.

Vuforia provides a number of different options for tracking, namely via 2D markers called VuMarks, 2D image targets, which may be any easily distinguishable 2D image, and 3D object targets. The creation and management of targets is mediated through the Vuforia Target Manager[21], which is available through the Vuforia developer portal. Before one can use the target manager, one is first required to register as a Vuforia Developer. A license key must then be created for each Vuforia application developed. Generating such license keys is free and can be done through the Vuforia License Manager[22]. Vuforia targets, which are stored in databases, can then be downloaded, via the Target Manager, for use in applications. Each database must be associated with one license key and one target database can contain different target types.

As the end-effectors are 3D objects, the 3D object target option was evaluated first. To make the relatively nondescript spherical end-effector more recognisable, a number of distinct custom monochromatic patterns were gen-

[20] `https://www.vuforia.com`

[21] `https://library.vuforia.com/articles/Training/Getting-Started-with-the-Vuforia-Target-Manager.html`

[22] `https://library.vuforia.com/articles/Training/Vuforia-License-Manager.html`

erated with Adobe Illustrator[23] and adhered to the spherical surface of the end-effector. The Vuforia Object Scanner application[24] was then installed on a Google Nexus 7 device and the end-effector was scanned while resting on a special Object Scanning Target provided by Vuforia. This target is printed on an A4 sheet of paper at its native size such that any physical scales estimated by the Object Scanner are relatively accurate. The Nexus 7's camera was slowly moved around the end-effector until enough reference points were captured for recognition. The scan was then saved as an Object Data file, allowing it to be stored in the target database. Once the 3D object target was

successfully stored in the target database, the database was downloaded and imported into the tangible hologram application through Unity. Databases are loaded through the `VuforiaConfiguration` asset via the inspector window in the Unity editor. Here, all datasets present in the application are visible and can be loaded and activated by clicking the appropriate checkboxes. The Vuforia SDK for Unity contains a number of different prefabs for setting up recognition and tracking targets, each one specific to the type of target in question. In this case an `ObjectTarget` prefab was placed on the stage and positioned at the origin. The prefab's properties are visible via the inspector when it is selected and a target, from a selected target database, can be associated with this prefab through the inspector. A Vuforia application also requires that an `ARCamera` prefab instance be present on the stage. A sphere GameObject with the same spherical radius as the end-effector was then added as a child object of the `ObjectTarget` prefab and positioned so as to align directly with the physical end-effector when rendered. The center of this rendered virtual sphere thus becomes the center of the physical end-effector in the tangible hologram application's coordinate space.

A `CustomTrackableEventHandler` script, which may be attached to any Vuforia tracking prefab, can be used to perform some action when Vuforia detects and tracks a target. The script sends the coordinates of the nested sphere GameObject to the `MainController` instance so that they can be processed further. Upon launching the tangible hologram application, it was noted that the quality of detection and tracking, when using the 3D object target, was limited, with a perceptible lag in detection and a significant amount of target loss when the end-effector was moving around. It was then decided to evaluate 2D image targets, which proved to be far more

[23] `http://www.adobe.com/products/illustrator.html`

[24] `https://library.vuforia.com/articles/Training/Vuforia-Object-Scanner-Users-Guide`

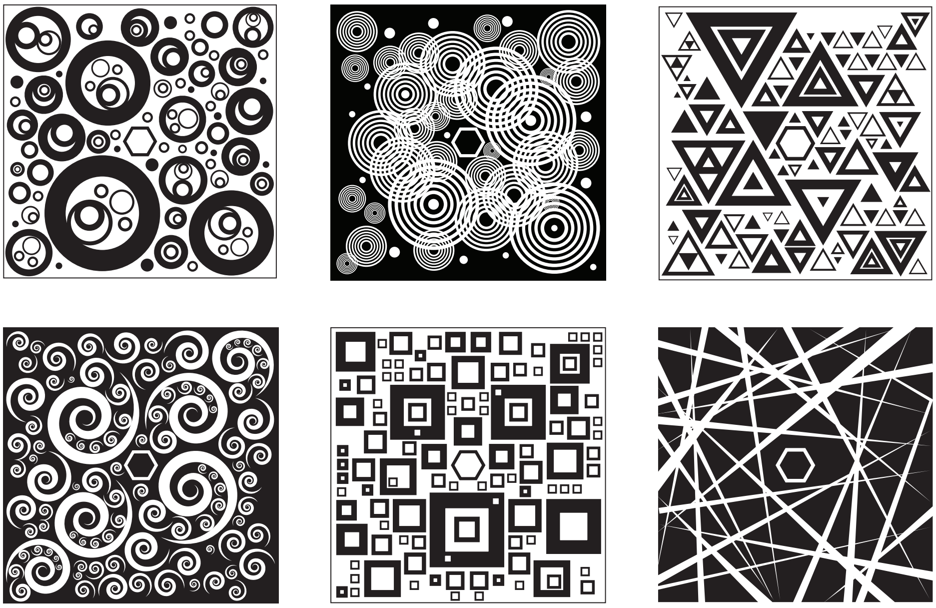


Figure 5.24: Custom Vuforia tracking image targets

performant. Two differently shaped end-effectors, one sphere with the same pattern applied to several areas of its spherical surface, and one cube, with a different pattern applied to each face, were trialled and a considerable improvement in detection was noted. In addition the targets performed well when the end-effectors moved about at speed with a marked reduction in the amount of times the target was lost. The custom patterns used for each face of the cuboid end-effector are shown in Figure 5.24

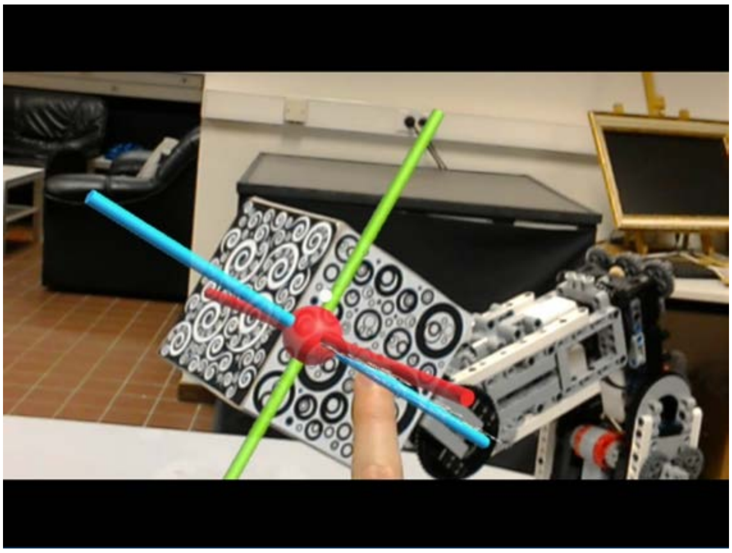


Figure 5.25: Vurforia end-effector tracking

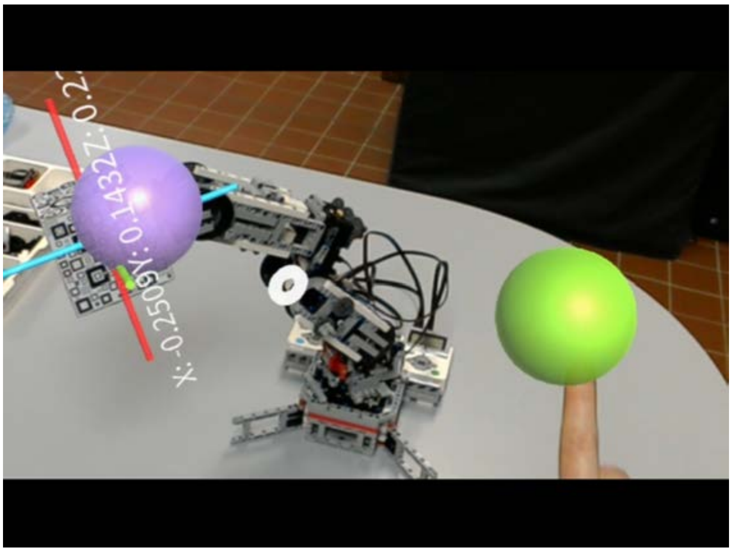
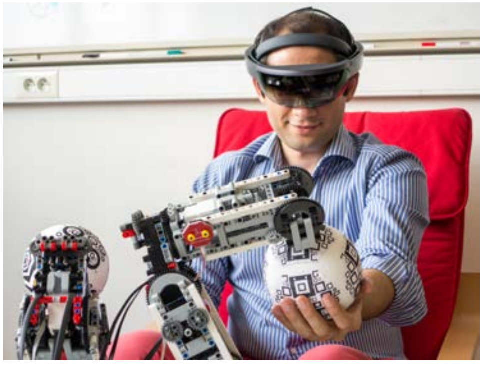
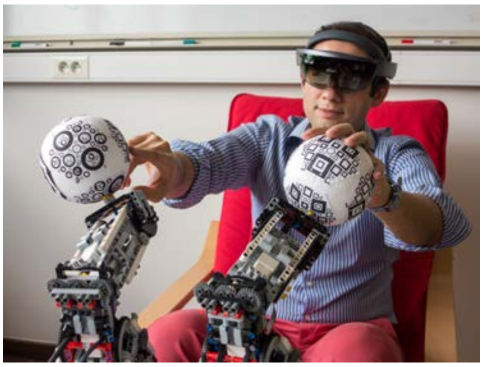


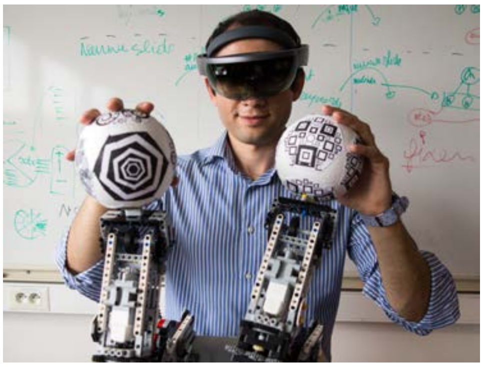
Figure 5.26: End-effector positioning



(a) Holding



(b) Poking



(c) Wall simulation

(d) Touching

Figure 5.27: Interactions with the TangHo system

6
Use Cases

In this section we present two use cases for the tangible hologram system and these should serve to illustrate TangHo's potential in a range of different scenarios. One of these use cases is a refinement of a similar scenario proposed by Signer and Curtin [42] in an earlier paper on the tangible hologram system and focuses on the employment of the platform in supporting an authoring environment for those working in creative design-related fields, such as automotive design. In the second use case, the art museum scenario, the system becomes a integral part of the museum experience, helping users to navigate its housed collections and providing rich and engaging ways to interact with various artworks. The use of the system in exploring and analysing world news content is illustrated in the final scenario and here the context of use, as well as the background and motivations of the users, can vary. The common thread, however, is that the scenario aims to demonstrate the benefits of representing the same data set in a multitude of ways, thereby leveraging the benefits of such an approach as set out in the *Affective Interactions with Data* section in the earlier background chapter. These use cases assume a more refined version of the system, coupled with the employment of a variety of end-effectors capable of handling complex user interactions with holographic content and generating complex output in return. The scenarios are, therefore, aspirational in nature, showcasing what might be possible with some further development of the tangible hologram system.

6.1 Art Museum Scenario

Three visitors to an art museum, an expert who has brought along two friends, are invited by the museum director to avail of the tangible hologram system in order that it might enhance their art exploration experience. The director takes them to an exploration room where three tangible hologram systems are made available to the group. This room is similar in function to those media rooms commonly found in museums and galleries, which are dedicated to the exploring the contents of the given institution, and themes related to them, through various media. Unlike such media rooms, however, this exploration room is Spartan in nature save for a large table in the centre of the room and a small book and postcard stand similar to those found in the museum shop. After receiving some instruction on how to use the tangible hologram system, the group begins. At this point the director explains that their augmented museum experience will consist of three distinct stages, a pre-visit stage, where the museum can be explored entirely through the tangible hologram system, a tour of the museum itself, again augmented by the system, and a post-visit stage that can be undertaken either in the museum or at home.

During the initial pre-visit stage, the group are encouraged to explore the art in the museum through the tangible hologram system, from within the aforementioned exploration room. The large physical table, situated in the centre of the room, provides a space from which to begin their exploration. Several hand-sized holographic spheres are situated around the table's periphery which may be imbued with the qualities of any desired material. These can then be manipulated by members of the group in a variety of ways. To begin their exploration, one of the participants decides to search the museum for a sculpture they recollect seeing at some point in the past. They are not sure of its name or the period but can remember the general form of the piece. They pick up a holographic sphere from the table and signify to the system, through a voice command, that they would like this sphere to behave as clay. The sphere now changes to a dark grey colour and inherits the plastic properties of clay. It is now possible for the group member to roughly model the form by manipulating the sphere as one would do with actual clay. The system supports their manipulations of the virtual sphere. As they apply pressure at certain points the holographic sphere deforms accordingly showing the results of their efforts. However, in addition to this visual information, they also experience the same haptic cues of resistance and friction as they would when working with the actual medium. Upon lifting the material off the table to manipulate the body as a whole, the user notices its weight. While modelling, they determine that the material is too resistive to their manipulation and specify to the system, through another voice command, that they wish the material to be more malleable. The system responds accordingly, changing the colour of the material to a lighter tone of grey to signify that it is now more pliant. Once they are happy that the form roughly resembles the piece in question they query the museum database by placing their modelled form on a specially designated part of the table. The museum system responds by displaying the results to the group member. A holographic matrix is visually rendered on the tabletop, whereby the columns divide the results into periods and the rows into geographic regions. The elevation of individual cells signify those results closest in form to the user's modelled version. The group member immediately recognises the piece they were searching for and they reach over to lift the piece off the matrix. This idea of querying by modelling may also be applicable if one wanted to investigate the manifestation of a particular figurative pose in different media and across various genres, for example.

At the other side of the table, another user, more knowledgeable in the field of fine art, wishes to explore the relationships between Vincent Van Gogh and other artists they collaborated with or met at some point. They make their way over to the small book and postcard stand and find a postcard of Van Gogh's self portrait. The system recognises the piece of work and offers a small holographic pot, filled with soil, to the user, who takes it in their hands and then place it in another part of the room. Van Gogh's portrait appears now as a label, resembling those found in potted plants, stuck into the soil in the pot. Once the pot is placed on the ground, a tree begins to grow with branches appearing for every artist Van Gogh had a connection with, with each artists name being carved into the bark. The location and thickness of the branch denotes the strength of the connection between the two artists, with those thickest branches closest to the root being the strongest. Once the tree is fully grown, *'fruit'*, in the form of transparent spheres containing works of art by Van Gogh, appear at the branch edges. The size and weight of this fruit determine the degree to which a given work of art has been inspired or influenced by the artist at the branch. The user wishes to compare two roughly equally sized spheres from different branches, and so they pick these items off the tree and hold one in each hand to determine if they are indeed equal. The tangible hologram system supports this form of unsupported holding and the arms exert the necessary downward pressure on the user's hands, via the end-effectors, to haptically render the weight. The user is satisfied that the two items are indeed approximate enough in weight to show that the two artists in question had a relative influence on these respective works by Van Gogh.

The final member of the group wishes to explore the impact a number of different movements had on the world stage and find out how much work is classified under each label. Like the previous group member, they make their way over to the book stand and select books on Art Nouveau, Impressionism, Post-Impressionism, Modernism and Surrealism. Returning to the large table, they arrange the books in front of them in an arbitrary order and then take a holographic globe, positioned nearby, and hold it in their hands. As they move the globe over each book, the globe responds by altering its contours, elevating countries, and areas within countries, relative to the impact that particular movement had in those locations. As they motion the globe from left to right and back again, they observe the growing and declining peaks, giving them an idea of the relative impact of these movements worldwide. The user then stops above the book on Art Nouveau and, while rubbing their hand over a large area of the globe facing them, detect a sharp narrow point. Looking closer at the globe they realise this is the city of Prague. As they were moving the globe over the different books, they sensed a noticeable change in the weight of the globe, this denoting the amount of work classified as belonging to each movement. Noting that Impressionism and Post-Impressionism are closely related, the user decides to combine these two by stacking the books one on top of the other. They ensure that the spines of the books are facing them so that the system can recognise them and continue on with their exploration. By stacking the books together in this way, the user can build queries relating to multiple art movements.

Happy with their initial explorations, the group now proceed through the gallery, each still wearing a tangible hologram system. A virtual tour guide gives an overview of important works in each room as they pass through. As they move into the wing housing the Van Gogh collection, one group member, who earlier carried out an exploration of Van Gogh's associations, notices a pulling sensation on their right hand, which rests on an end-effector. As they walk on, the pulling sensation becomes more intense, almost behaving as a magnet of sorts. Eventually they decide to follow the direction of the pulling movement, which stops as they arrive at a painting they earlier encountered in their explorations. The painting is *The Starry Night* and they pause in front of it for a moment to observe it. The textures in the painting are rich and they wonder what it would be like to reach out and touch it. Standing back a distance from the painting, they indicate, with a voice command, that they would like touch the piece. The tangible hologram system responds by generating a holographic version of the work, which sits between them and the original. As they make contact with the holographic painting, they perceive that the textures are faithfully rendered and, through their haptic explorations of the painting's surface, they get an intuitive sense of how the paint was applied. They can almost feel the energy within the work through the haptic perception of the paint strokes.

Once their tour is concluded, the group return to the exploration room and there they briefly discuss their experiences with the tangible hologram system. The director offers them those postcards and books they interacted with earlier so that they can relive their museum experience at home. All their earlier interactions within the museum have been recorded and are now uploaded to an online data store so that they might be accessed later in a different context.Those postcards, books and individual images within books that the group interacted with now serve as windows through which they can relive their museum experience. Later on that evening one group member decides to review their interactions with the system at the museum. They put on their personal tangible hologram system and place a number of books from the museum out in front of them. The familiar globe appears at their side. They pick it up and carry on where they left off earlier.

6.2 Automotive Designers Scenario

This scenario was first introduced by Signer and Curtin [42] will be discussed and expanded upon here. It is customary for those designers working in the automotive industry to use plasticine clay to model conceptual car designs at a range of scales, including full scale models. Indeed, the practice of using plasticine clay in this fashion has persisted for decades and is still employed by many automakers at present, in spite of recent advances in 3D authoring software. Ford, for example, uses about 100 tons of the material in a given year[1]. One advantage of the approach is that it allows for collaborative modelling, especially at larger scales and has that *always on* property—cited by Jansen et al. [23] as an inherent feature of data physicalisations important in supporting *"casual visualisation"*. A more intuitive understanding of lines of the car can be fostered from studying such scale models, as one can walk around them to get a full sense of their form and study them up close simply by moving closer to them. It is also much easier to gauge if features are suitably proportioned when one can actually perceive a full-sized model manifested in the physical world. Augmented reality headsets can now project virtual models, at any scale, into the physical world, thus realising many of the benefits of scale models just discussed. However, it is the production of such models that is the central concern here.

[1] `http://www.bbc.com/autos/story/20161111-why-car-designers-stick-with-clay`

The aforementioned plasticine clay offers many advantages over computer-mediated authoring approaches. One major advantage is that it is quite easy to work with and interactions with the material are almost instinctive, leveraging natural spatial and haptic perceptual abilities. If one makes a mistake or wishes to modify a certain part of the model one can do so through direct interaction with, and manipulation of, the medium. Input and output are bound as one and there is no layer of indirection. Large full-scale models are built over an armature of light aluminium with adjustable fittings representing those fixed elements of the design as set out in the project brief[1]. Smaller models may also use an armature and can be roughed up in a brisk fashion. If a given rough model is worthy of further development, it can be worked up to a higher degree of refinement. Should features of a refined model require simplification, this, too, is easily achievable. Plasticine clay comes in a variety of stiffness's and may be used to model features to high level of precision, with stiffer material allowing for more precise work. Another advantageous property is that it the clay does not harden upon exposure to air and so rendered designs and features are easily customisable at any stage of the process. Modelling with the medium affords additive and subtractive processes and one can use a wide range of tools to render various features and details. Indeed, the material was used to model the base sections of the tangible hologram system with good results.

Given the benefits of plasticine clay just discussed, it is clear to see why the practice of modelling with such clay is still so popular. Any system hoping to provide an alternate approach would need to emulate the material properties of plasticine clay faithfully. Indeed, to ensure the adoption of an alternative to clay, one would need to offer more than just its high-fidelity simulation, although this alone would generate significant financial savings. The real gains come from the virtualisation of the entire modelling process, a virtualisation that preserves and enhances the practice of working with physical clay. Virtualisation also negates the need to translate the physical into the digital and visa-versa, as only one form of the model exists—a physically augmented virtual model. There would also be no need to soften those stiffer clay varieties by pre-heating them and the relative plasticity of the virtual material could be adjusted on the fly. To illustrate how this might work, the original scenario will now be introduced and expanded upon.

An automotive designer is working, through the tangible hologram system, on a holographic clay model of a concept car in the early stages of development. The designer works at a bench in their office and the rendered model is small in scale. Similar to the interaction with holographic clay discussed in the art museum scenario, the designer effects changes through direct hands-on manipulation of the holographic model. They see the results of their manipulations reflected in the hologram and can sense those same changes physically through the haptic rendering. As before, the system is capable of rendering the plastic quality of the clay through a sophisticated end-effector and the designer gets the same haptic cues of resistance and friction mentioned before. As this model is in its early stages the designer works quickly and the virtual clay yields easily to the touch. After an hour or so the designer begins to refine some of the details and increases the resistance of the material so as to allow for more precision. They also scale the model up a little to work on some of the details in greater detail. The system now binds this larger scale higher-resolution to the original smaller scaled model. After working up some of the details the designer feels that they are not entirely happy with the shape of the car's bonnet. They place the larger scale model to one side and bring up the smaller model. With a few simple manipulations they reshape the bonnet and these coarse changes are reflected and accommodated into the larger model, taking into account its scale. Some time later the designer, happy with their work so far, decides to show the model to their colleague, working in the next office, to get their opinion. The designer picks up the model and places it in the centre of the studio's common area. They then scale the model up to full size, with this scaled model bound to the previous two, and call their colleague out to get a second opinion.

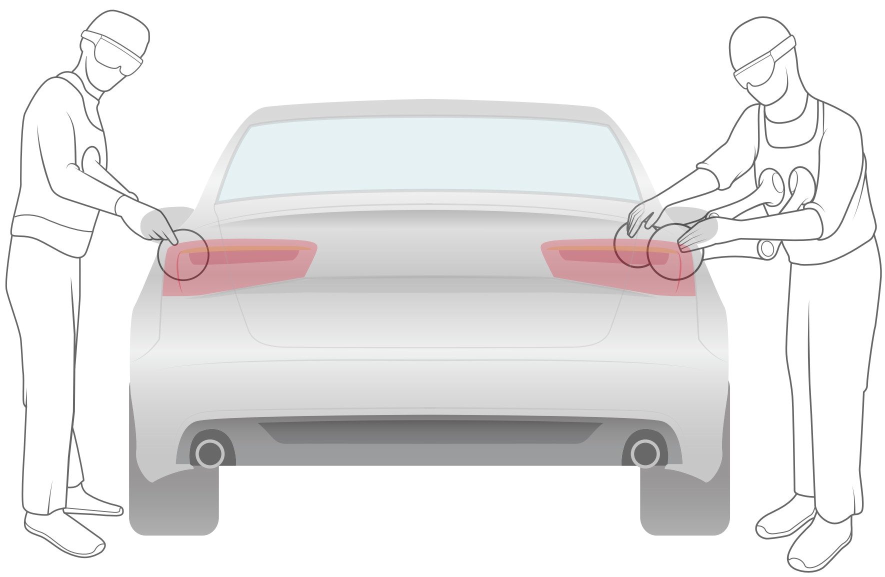


Figure 6.1: Automotive designers collaborating on a tangible holographic car model

The colleague, who is also wearing a tangible hologram system makes some observations and notes a couple of possible improvements. They both work on the full scale model, through the system, for a time, after which the original designer scales the model down to a reasonable size. This model, though smaller in scale is more detailed than if it were originally modelled at this scale. As such, this model, though it may be identical in scale to another, is bound to the design chain as a separate entity. As the project deadline is close at hand, the colleague offers to do some work on the model when they have some spare time and so get a copy of the design chain from the original designer. They can now both work on any iteration in the chain in tandem, with updates by one designer being reflected in the working models of the other—if this behaviour is desired. After the deadline has passed, the studio's designers have their usual retrospective meeting and the project in question is up for discussion. Only some designers are wearing a tangible hologram system, while the others can watch a live feed on the boardroom's television. Additionally a number of board members watch remotely with one member having access to a tangible hologram system. The group listens as the designer gives a presentation of their progression through the design process, aided by the system, from the early iterations through the final proposal that successfully satisfied the requirements of the brief. To help illustrate their point and generate some discussion around the design process, the designer lays out all the models comprising the design chain, in a chronological sequence, to that the others may see them. After the presentation, those wearing the system, including the remote participant, can pick up these models and have a closer look. At this point the models have been *locked* so they can be damaged or manipulated further. They now serve only as a record of the work undertaken.

6.3 World News Analysis Scenario

An analyst wishes to explore some recent developments in world news to prepare for a conference on global trends they will be attending the following week. The conference is on global inequality and the analyst wishes to make a meaningful connection with the data, so as to get a better understanding of the material. The analyst is working from home today and the company has given them a tangible hologram system to use in their exploration. Initially they take a look through a world newspaper that aggregates stories from all over the world. This is holographically manifested as a regular newspaper the analyst can pick up off their breakfast table and flick through as one would with a regular paper. As the analyst goes through some of the main stories of the day, their attention is drawn to a story on growing world hunger levels. This story is of interest and so the analyst tears the story from the holographic paper and pins it to a holographic corkboard, positioned nearby. The corkboard begins to sift through keywords contained within the article and offers a collection of related datasets, represented as hand-sized circular tokens. The analyst spots a data set relating to the amount of calories consumed per head of population and selects this data set by plucking it from the corkboard.

The data set is initially presented as a series of values projected onto the wall adjacent to the corkboard, and the analyst finds this representation of the data dense and hard to get a sense of. The analyst decides to display a world map on the floor of their living room and, while holding the token in their hands, begin to walk slowly over the map. As the analyst passes over different locations the token changes weight, as the relative weight of the token has been encoded to match the calorie consumption levels in the data set. The calorie consumption data is represented in such a way that the rendered weight of the token equates to that of an average meal for a person in the location over with the token currently finds itself. This helps the analyst to get a more intuitive sense of what the figures in the dataset mean, in human terms, as the data has now been rendered more relatable. The complexity of the initial data the analyst was presented with has been reduced to the scale of one person. Moving over Western Europe the analyst notes the sizeable weight of the token, however, as they move further south, toward more disadvantaged areas, they notice that the token weighs less and less—until eventually it weighs very little at all. This has an immediate impact on the analyst. The surprise at how little some people actually have to eat motivates them to find ways of expressing this data so that this type of inequality is made more explicit to a wide audience.

Later, the analyst comes across another news story about the amount of waste per country in a given year. Again, this piques their interest and, having had a powerful experience with the token previously, they initiate a similar scenario with the world map. Again the complexity of the data has been reduced to the scale of the individual. During this exploration the analyst notices that the weight encodings almost seem to have reversed and this gives the them pause for thought. The analyst begin to work on their final presentation for the conference, emboldened by the experience they have just had with the data—and determined to get the message contained within it to as many people as possible.

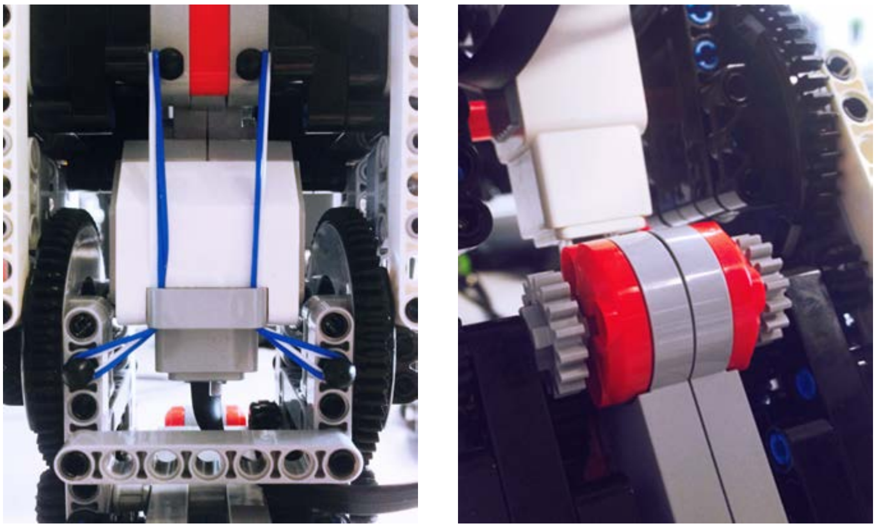
7
Future Work and Conclusion

The future work section highlights those areas of the tangible hologram system that merit further refinement and development. As it stands, the TangHo system can be viewed as a work in progress, with plenty of research potential. The section highlights some current limitations with certain aspects of this initial iteration of the system and suggests how one might begin to realise improvements.

7.1 Future Work

The tangible hologram system implemented here was the result of an exploration into how one might begin to develop such a platform with existing technologies. It should be viewed as a starting point from which others can learn, allowing them to and draw lessons from what was implemented and to reflect on the approach undertaken, so that, in time, they may refine the system and add to its functionality. Unfortunately, owing to time constraints, not all of the ideas and functionalities possible were implemented. The multifaceted nature of the project should provide many spaces in which one could substantially apply concepts from, and contribute further research to, a variety of fields, from robotics and kinematics, through to sensors, digital signal processing and embedded systems. Much too can be explored and developed in terms of designing rich haptically augmented holographic applications and the engaging interactions demanded by such applications. These applications can focus on the exploration and analysis of data, the intuitive authoring of creative content, or something else entirely. Indeed the platform has plenty of scope in terms of potential applications.

TangHo's robotic arms are, for the most part, built entirely from Lego Mindstorms components. This is advantageous in that it affords easy modification and adjustment. Gear trains can be reconfigured to provide more torque or speed as needed and the arms' modular design means that a refined or reimplemented section can still be compatible with the existing construction. In a similar fashion, 3D printed components or casings may also be designed in such a way so as to interoperate with elements of the Lego Mindstorms system and, by extension, existing parts of the arms themselves. The 3D printed base of our tangible hologram system shows that the combination of custom designed components and Mindstorms elements is achievable and delivers good results. The incorporation fo 3D printed casings may serve to lighten the structure of the tangible hologram system while still maintaining a high degree of robustness. Thus, the design and fabrication of such components is desirable and may be undertaken as future work.



(a) Shoulder joint elastic loops (b) Elbow joint

Figure 7.1: Joint implementation

The existing arm design and construction present a number of areas in need of further development. As mentioned previously, there is a trade-off between torque and speed when designing gear trains. As some of the arms' servo motors need to move more than a modest amount of weight, the gear trains attached to these motors have favoured torque over speed. This is reasonable when the arms are being driven under power. However, when one wishes to move the arm freely, to move a holographic object with the arm, for example, there is a certain degree of inherent resistance and this is undesirable. This effect is particularly noticeable at the arms' elbow joint. An effective strategy has already been employed to the arms' shoulder joint to help balance the opposing concerns of speed and torque. Elastic loops are used to provide some counter-resistance to the weight at this joint and this approach also allows the shoulder to support the weight of the rest of the arm even when the arm is not powered on. Thus the gear train driving the shoulder joint does not need to provide the same amount of mechanical advantage, thereby allowing for a less resistive configuration of gears. The shoulder joint is, therefore, freely rotatable when user-generated force is applied, while still supporting the majority of the arm's weight. An elimination of the intrinsic resistive forces present in joints such as the elbow would greatly improve the user experience when performing certain interactions. Ideally a user should not detect any unintended friction from the motion of the robotic arms' gears and an investigation into the use of smoother low-friction capstan drives, as a replacement for the current gear trains, may be fruitful here. It may also be possible to implement capstan drives using the existing servos and other components from the Lego Mindstorms system.

One area that has much potential for further development is the design of future end-effectors for the tangible hologram system. Presently the end-effectors are static spheres and cubes, which one can consider as placeholders for what might be developed later. As mentioned in earlier sections, the ultimate goal of TangHo is to haptically augment holograms with a range of different physical properties including temperature, texture and density. Additionally, in order to support the use cases discussed earlier, end-effectors, capable of affording a complex range of interactions, along with sophisticated haptic rendering, will need to be devised and implemented. These end-effectors may be as intricate in construction as the system's robotic arms, if not more so, and may involve the combination of a number of different haptic rendering approaches. One could imagine, for example, a more compact form of the TRANSFORM shape display, housed in an end-effector sitting at the end of each robotic arm. Such a combination would grant the mobile fine-grained rendering of complex forms, thus opening up a wide range of possibilities for designing engaging interactions. Another possible end-effector design might involve an actuated system encased in a deformable or elastic material. Here, one can imagine an arrangement of force sensing linear actuators originating from a central point, so that a spherical shape is formed when all are fully contracted. This arrangement is covered with the deformable or elastic material so that the user is presented with a perfect sphere. When the end-effector is required to assume the form of a nearby hologram, each linear actuator extends the necessary distance to match a given point on the virtual object's surface. Should the user wish to manipulate the form of a virtual object, they can do so through the direct manipulation of the end effector. Pressure generated by the user's manipulating forces can be sensed by force sensing actuators, which, in turn, can relay this information back to the system. The system can respond by adjusting the extension of each actuator's protrusion accordingly. In this way materials with varying degrees of stiffness may be rendered. Evidently, these concepts are not trivial to realise, however, this makes the design and implementation of such end-effectors an exciting challenge.

Hand and finger tracking with the HoloLens is another area that merits further exploration. At present, access to hand tracking information is limited to the position, without orientation, of the centre of the user's hand. In addition, the hand has to be either the *ready state*, where the index finger points upwards and the back of the hand faces the user, or the *pressed state*, where the index finger points down. The HoloLens will simply ignore hands that are not in these states[1]. Hand tracking for the tangible hologram system is limited to tracking the hand when it is in the ready state. While this is sufficient to some develop and test some initial interactions and get a sense of what might be possible, the system will require more involved hand tracking in future iterations. It is apparent that the HoloLens is capable of recognising the location and position of individual fingers, in order to perform gesture recognition, however, developers do not have access to this information as it stands. A number of different approaches might be applicable in order to push the envelope here. One approach might employ the use of a hand tracking device such as leap motion[2], which some developers have successfully coupled to the HoloLens[3,4]. Adopting computer vision techniques may also yield some strong results and computer vision libraries, such as OpenCV, might prove beneficial here.

In addition to hand tracking, the tracking of the system's end-effectors also warrants deeper investigation. At moment the system uses the Vuforia software platform to perform the recognition of the end-effectors, using custom tracking image targets in the process. While this solution is satisfactory for static end-effectors with simple geometric form, such as cubes and spheres, dynamic variants with more complex forms may be harder to

[1]`https://developer.microsoft.com/en-us/windows/mixed-reality/gestures`
[2]`https://www.leapmotion.com`
[3]https://www.youtube.com/watch?v=HqFZKViMUCg
[4]https://www.youtube.com/watch?v=0Rso0nxzfQU

(a) © Leap Motion (b) © Leap Motion

Figure 7.2: Leap Motion - (a) mounted to a VR headset and (b) example application

track with in this way. Vuforia does offer the ability to scan 3D objects so that the system can recognise them later and this was trialled during the development of the tangible hologram system. However, a perceptible lag in achieving initial recognition was noticed with this approach. Moreover, when the end-effectors moved around at speed, the tracking was temporarily lost. This initial lag and loss of tracking were not nearly as pronounced with the image target method. It must be noted that Vuforia have only recently started to work with HoloLens and their 3D object recognition and tracking feature might well improve in time, however, this offering does not meet the demands of the tangible hologram system at the moment. In the short term, other object recognition approaches may need to be sourced and evaluated to see what they can offer and how they might be used to complement or replace what is currently in place.

As a general note on what might be pursued in future work, it is worth highlighting that the system has a lot of moving parts and components, each of which can be modified, replaced or refined. For example, the kinematics solvers found in Peter Corke's robotics toolbox have been written with a pedagogical focus. This is highly beneficial for developing an understanding of how the code itself works and, as previously mentioned, the toolbox code is referenced in Corke's introductory textbook to robotics, thereby allowing one to follow and appreciate the transition from theory to implementation. The kinematic solvers employed by the tangible hologram system, then, may be refined or reimplemented to achieve performance gains, where it is possible to do so. Indeed, this would constitute an important improvement as any performance optimisations will have a direct bearing on the perceived responsiveness of the system. As tangible hologram applications become more refined, and the computational complexity associated with such applications increases, staying within the 100 ms bound, important for maintaining the perception of good system responsiveness, becomes more challenging. Therefore any improvements in performance elsewhere in the system are important contributions.

In terms of the capabilities of the robotic arms themselves, there is much that can be refined and developed. User experience would benefit greatly from smoother arm dynamics, where movements are more fluid and less sudden. Improvements here would create a less jarring experience when users reach to touch holograms, while also generating less intrusive forces on those parts of the body to which the device is grounded. Another important additional capability, essential for many interactions, is force feedback rendering. At the moment the arms move the end-effectors so as align their surfaces with that of holograms, in order that they might be touched and grasped. However, the degree of force reflected back on the user's hand is not currently treated and this will require remedying. Indeed, force feedback may need to be realised in both dynamic actuated end-effectors and the robotic arms themselves. This requirement may place some unreasonable demands on the existing Lego Mindstorms construction and further investigation will need to be undertaken to evaluate how suitable the current servo motors may be for this purpose.

One of the largest areas for future work is in designing and implementing applications that fully utilise the current and future potential of the tangible hologram system. Some possible applications have been discussed in the use case section to illustrate how the capabilities of the system may be suitably harnessed. User interaction with the content of applications will need careful design so that one's haptic and spatial perception abilities are effectively engaged, thereby ensuring that the benefits of natural and intuitive interaction are realised. Ideally, such applications would levearge the inherent capabilities of the HoloLens as much as possible. For example, good use can be made of the HoloLens' stereo sound to enhance haptic cues. If one knocks on a hologram representing a hollow object, one should both feel the haptic sensation on their finger and hear an echoing sound emanating from the the hologram itself. In addition directional sound can be very useful when the application involves the exploration of a data-driven environment, providing potentially important non-visual and non-haptic cues. Good use should also be made of the device's voice and gesture recognition features, where applicable. In terms of supporting data exploration and analysis, applications might make use of multiple representations of the same data set, with some representations affording affective interactions with the data, while other representations may focus on making the information, and underlying patterns and relationships therewithin, explicit and easily comprehensible.

Unfortunately, owing to time constraints, we were unable to carry out a full user study of the tangible hologram system and the undertaking of studies such as this will be an important future contribution to the project. Here the TangHo system should be evaluated in terms of usability and the quality of interactions it affords. An interesting study might compare its efficacy in facilitating the exploration and analysis of data with existing data physicalisation approaches. Another study might focus on how effective the system is in haptically rendering physical properties, such as density, weight and inertia. The qualitative data gathered from such a study would be beneficial gaining an insight into how detailed such haptic rendering needs to be. A comprehensive quantitative evaluation of the system's responsiveness and accuracy will also need to be undertaken, highlighting where optimisations should be made and where estimations could be improved.

7.2 Conclusion

The TangHo system represents the breaking of new ground in the area of tangible hologram systems and may serve as a useful starting point for those wishing to work on the research and development of such systems. It constitutes the synthesis of a number of different technologies and the application of various concepts, which work in tandem to realise the novel idea of the mobile physical augmentation of holographic content. Those who wish to further the work and continue the endeavour of actualising such tangible hologram systems might refer to the approaches adopted in this implementation and draw from the lessons learned in the undertaking.

The design and development of the TangHo system were initiated out of a desire to address some of the implementation and interaction challenges associated with the physicalisation of data. From an initial investigation into the suitability of Tangible User Interfaces in aiding data exploration and analysis, the programmable matter of Ishii's Radical Atoms vision came to light as a powerful concept meriting further investigation. Its beneficial application to the area of data physicalisation, were it to be realised, prompted us to question if it might be possible to simulate the experience of working with the material using current technologies. The TangHo system presented in this thesis thus demonstrates how the promise of programmable matter might be approximated so that users can begin to prototype and evaluate new possibilities in terms of physical data interactions today, rather than tomorrow. As such, the system can be viewed as a platform for research into the design of new dynamic materials and the interactions they might afford.

A substantial portion of the work undertaken in this thesis has seen an exploration into those areas worthy of consideration when designing and implementing a tangible hologram system. The discussion of salient topics and themes central to these areas, as presented in the background section, might prove useful as a guide through the areas of data physicalisation, haptics, TUIs and programmable matter, in terms of how they relate to each other and how they relate to the area of tangible hologram systems. This investigation and the discussion arising from it can be regarded as a contribution of this thesis.

It can be said that the choice of components used to build the tangible hologram system yielded many advantages, and, in some cases, presented a number of shortcomings. The decision to employ the Lego Mindstorms system in the construction of the haptic unit prototype, proved, in many ways, to be a beneficial one. The design and implementation of the system's arms, while at times an involved and lengthy process, benefited greatly from the forgiving nature of the Mindstorms system and modifications and reconfigurations were not as taxing as they might have been with other approaches. Thus, the Mindstorms system allowed for a high degree of experimentation and facilitated the iterative development of the prototype. It was planned, from the outset of development, that the arms' mechanisms would eventually be housed in custom 3D printed components, thereby eliminating those parts of the Mindstorms construction constituting the robots' frames and retaining just the servo motors, gears and axles. However, due to time constraints, only those base components of the system were fabricated and the remaining Mindstorms elements interoperate with these. This is not an undesirable outcome in that a completely sealed system may, in many respects, hinder the ability to modify, refine and reconfigure the current iteration of the robotic arms. The TangHo system is a research platform in its infancy and, as such, every part of it should be open and accessible to those wishing to contribute more to the idea. The digital model of the Mindstorm construction, produced as part of this thesis, helps to extend this accessibility. It must also be noted that final construction is relatively robust, enough so to qualify it as a suitable prototype for the TangHo system's haptic component.

The Microsoft HoloLens also displays many positive qualities in its role as TangHo's mixed reality element. It offers currently unparalleled inside-out positional and rotational tracking, along with a high degree of hologram stability. The holograms generated by the device are convincingly grounded in the physical world and interact with it accordingly. Most importantly for the TangHo system, the HoloLens is an untethered device, granting unimpeded mobility and allowing for an unlimited working area, thus increasing the possibilities for applications developed with the TangHo platform. However, as mentioned previously, its restricted access to hand tracking information and its limited field of view are distinct drawbacks, having some noticeable effect on the quality of interactions and user experience. As discussed in the future work section, there may be ways to circumvent the hand-tracking restrictions, however, the limited field of view will only see improvement in future iterations of the HoloLens headset.

One of the most challenging aspects of implementing the tangible hologram system was the need to support mobility. This requirement placed significant demands on the design of the haptic system and ruled out the possibility of adopting more straightforward approaches in a number of areas. A mobile haptic unit necessitated the use of IMU sensors, for example, and working with IMUs, and the processes required to accurately determine sensor orientation, added another layer of complexity to the project. That being said, mobility constitutes an important feature of the TangHo system and, as such, the additional time and effort required to realise this capability was given its due allocation in the project.

The limitations of this initial iteration of the tangible hologram system have been discussed in the future work section and highlight that the system is a work in progress. As such, the TangHo project is rich in potential areas of research that will serve to further the platform and grow its capabilities. As it stands, the system represents a starting point for the development of systems affording the mobile haptic augmentation of holographic content—systems that put the potential of programmable matter into the hands of those endeavouring to develop data physicalisations and the rich interactions that go with them.

A
Appendix

A.1 Notable Java Classes

Listing A.1: SensorReader class for reading and adjusting IMU data

```
package sensors;

import java.util.HashMap;
import lejos.hardware.Button;
import lejos.hardware.port.Port;
import lejos.hardware.sensor.SensorMode;

public class SensorReader {
 // Default values for Hz etc.
 private static int samplingHz = 50;
 private static int offsetCalculationIters = 200;
 public static enum SensorTag { BASE, LEFT_1, LEFT_2, RIGHT_1, RIGHT_2 };
 private static HashMap<SensorTag, SensorOffsets> sensorOffsetsMap;
 private static boolean sensorDataInitialized = false;

 //private MindsensorsAbsoluteIMU imu;
 private MindSensorsAbsoluteIMUMod imu;
 private SensorTag sensorTag;
 private SensorMode accelSensor;
 private SensorMode gyroSensor;
 private SensorMode magSensor;

 private float[] accelSampler = new float[3];
 private float[] gyroSampler = new float[3];
 private float[] magSampler = new float[3];
 // Defaults for individual sensors will be loaded upon initialization
```

```
 // of sensor data
 private SensorOffsets offsets;
 private float[] accelOffsets;
 private float[] gyroOffsets;
 private float[] magOffsets;
 private float[] magScales;

 // readValues is the array of values to return
 private float[] readValues = new float[9];
 // sampleMillis for calibration routines
 private int sampleMillis = 1000 / samplingHz;
 private boolean recalibrated = false;

public SensorReader(Port port, SensorTag tag) {
 if(!sensorDataInitialized) {
 initializeSensorData();
 }
 //imu = new MindsensorsAbsoluteIMU(port);
 imu = new MindSensorsAbsoluteIMUMod(port);
 sensorTag = tag;
 accelSensor = imu.getAccelerationMode();
 gyroSensor = imu.getRateMode();
 magSensor = imu.getMagneticMode();

 // Get preloaded offsets
 offsets = sensorOffsetsMap.get(sensorTag);
 accelOffsets = offsets.getAccelOffsets();
 gyroOffsets = offsets.getGyroOffsets();
 magOffsets = offsets.getMagOffsets();
 magScales = offsets.getMagScales();
}

private class SensorOffsets {
 private float[] aOffsets;
 private float[] gOffsets;
 private float[] mOffsets;
 private float[] mScales;

public SensorOffsets(float[] aOff, float[] gOff, float[] mOff, float[]
    mScale) {
 aOffsets = aOff;
 gOffsets = gOff;
 mOffsets = mOff;
 mScales = mScale;
}

public float[] getAccelOffsets() {
 return aOffsets;
}

public float[] getGyroOffsets () {
 return gOffsets;
}

public float[] getMagOffsets() {
 return mOffsets;
}

public float[] getMagScales() {
 return mScales;
}
```

```
public float[] read() {
 accelSensor.fetchSample(accelSampler, 0);
 gyroSensor.fetchSample(gyroSampler, 0);
 magSensor.fetchSample(magSampler, 0);

 readValues[0] = (gyroSampler[0] - gyroOffsets[0]);
 readValues[1] = (gyroSampler[1] - gyroOffsets[1]);
 readValues[2] = (gyroSampler[2] - gyroOffsets[2]);
 readValues[3] = (accelSampler[0] - accelOffsets[0]);
 readValues[4] = (accelSampler[1] - accelOffsets[1]);
 readValues[5] = (accelSampler[2] - (accelOffsets[2] - 1.0f));
 readValues[6] = (magSampler[0] - magOffsets[0]) * magScales[0];
 readValues[7] = (magSampler[1] - magOffsets[1]) * magScales[1];
 readValues[8] = (magSampler[2] - magOffsets[2]) * magScales[2];

 return readValues;
}

public void recalibrate() {
 accelOffsets = SensorUtils.getMeanValues(accelSensor,
     offsetCalculationIters, sampleMillis);
 gyroOffsets = SensorUtils.getMeanValues(gyroSensor, offsetCalculationIters,
      sampleMillis);
 recalibrateMagnetometer();
}

private void initializeSensorData() {
 // Values recorded from each sensor
 // AccelOffsets, GyroOffsets, MagOffsets, MagScales
 sensorOffsetsMap = new HashMap<SensorTag, SensorOffsets>();
 // Still need to do proper magnetometer values
 sensorOffsetsMap.put(SensorTag.BASE, new SensorOffsets(
 new float[] {-0.17294657f, 0.06566822f, 0.9780375f},
 new float[] { 0.37152982f, 0.2642045f, 0.45612603f},
 new float[] {-0.08790909f, 0.2162273f, -0.91576517f},
 new float[] {0.9278408f, 1.0738621f, 1.3175104f}
));


 // Still need to do proper magnetometer values
 sensorOffsetsMap.put(SensorTag.RIGHT_1, new SensorOffsets(
 new float[] {0.058646612f, -0.008212778f, 0.9727149f},
 new float[] {0.020984245f, -0.05367776f, 0.8532404f},
 new float[] {-0.43227276f, -0.1434091f, -0.13137753f},
 new float[] {2.0164123f, 1.0808436f, 0.65766925f}
 ));
 sensorDataInitialized = true;
 System.out.println("SensorDataInitialized");
}

private void recalibrateMagnetometer() {
 System.out.println("Calibrating Magnetometer - Please press the Enter
     button to begin");
 Button.ENTER.waitForPressAndRelease();

 // Hard Iron Calibration
 int magCalibrationSamples = 250;
 float[] magCaliXVals = new float[magCalibrationSamples];
 float[] magCaliYVals = new float[magCalibrationSamples];
 float[] magCaliZVals = new float[magCalibrationSamples];
```

```
  for (int i = 0; i < magCalibrationSamples; i++) {
   magSensor.fetchSample(magSampler, 0);
   magCaliXVals[i] = magSampler[0];
   magCaliYVals[i] = magSampler[1];
   magCaliZVals[i] = magSampler[2];
   try {
    Thread.sleep(sampleMillis);
   } catch (InterruptedException e) {
    e.printStackTrace();
  }
 }

  float[] mXminMax = SensorUtils.getMinMax(magCaliXVals);
  float[] mYminMax = SensorUtils.getMinMax(magCaliYVals);
  float[] mZminMax = SensorUtils.getMinMax(magCaliZVals);

  magOffsets[0] = (mXminMax[1] + mXminMax[0]) / 2.0f;
  magOffsets[1] = (mYminMax[1] + mYminMax[0]) / 2.0f;
  magOffsets[2] = (mZminMax[1] + mZminMax[0]) / 2.0f;

  // Soft Iron Calibration
  magScales[0]  = (mXminMax[1] - mXminMax[0]) / 2.0f;
  magScales[1]  = (mYminMax[1] - mYminMax[0]) / 2.0f;
  magScales[2]  = (mZminMax[1] - mZminMax[0]) / 2.0f;

  float avg_rad = magScales[0] + magScales[1] + magScales[2];
  avg_rad /= 3.0f;

  magScales[0] = avg_rad/magScales[0];
  magScales[1] = avg_rad/magScales[1];
  magScales[2] = avg_rad/magScales[2];

 }

 public SensorTag getSensorTag() {
  return sensorTag;
 }

 public void setSensorTag(SensorTag tag) {
  sensorTag = tag;
 }

 public boolean isRecalibrated() {
  return recalibrated;
 }

 // static methods for getting / setting the sampling frequency etc
 public static int getSamplingFrequency() {
  return samplingHz;
 }

 public static void setSamplingFrequency(int frequency) {
  samplingHz = frequency;
 }

 public static int getOffsetCalculationIters() {
  return offsetCalculationIters;
 }

 public static void setOffsetCalculationIters(int iters) {
  offsetCalculationIters = iters;
 }
```

```
}
```

Listing A.2: MatlabRobot class for handling communication with the MATLAB engine

```
package armApplication;

import java.io.StringWriter;
import java.util.Arrays;
import java.util.concurrent.CancellationException;
import java.util.concurrent.ExecutionException;
import com.mathworks.engine.*;

public class MatlabRobot {

 private MatlabEngine mEngine;
 private StringWriter mOutput;

public MatlabRobot() {
 mOutput = new StringWriter();
}

public boolean startRobot() throws Exception {
 mEngine = MatlabEngine.startMatlab();
 mEngine.eval("robot_arm = SixAxisRobot", null, mOutput);
 mEngine.eval("robot_arm.name = 'ARM1'", null, mOutput);
 return engineRunning();
}

public double[] getAngles (double[] currentJointAngles, double[]
    hololensEffectorCoords, double[] moveToCoords) {
 double[] solution = null;
 runMatlabCommand("ikSolution = robot_arm.solveIK(" + Arrays.toString(
     currentJointAngles) + ", " +
 Arrays.toString(hololensEffectorCoords) + ", " + Arrays.toString(
     moveToCoords) + ")");
 try {
 solution = mEngine.getVariable("ikSolution");
 } catch (IllegalStateException | InterruptedException | ExecutionException
     e) {
 e.printStackTrace();
 }
 return solution;
}


// Make this string easier to construct as it might be slowing things down
public void setRobotBase(double[] angles, double[] coords) {
 runMatlabCommand("robot_arm.setBase(rt2tr(rpy2r(" + angles[0] + ", " +
     angles[1] + ", " +
 angles[2] + ", 'deg'), " + Arrays.toString(coords) + "'));");
}

public void plotRobotPosition() {
 runMatlabCommand("robot_arm.plotCurrentPos");
}

public void runMatlabCommand(String command) {
 try {
  mEngine.eval(command, null, mOutput);
```

```
 } catch (CancellationException | InterruptedException | ExecutionException
     e) {
  e.printStackTrace();
  System.out.println("MATLAB error redirected to Java: ");
  System.out.println(mOutput.toString());
 }
}

public boolean engineRunning() {
 return mEngine == null ? false : true;
}

}
```

A.2 Class Diagrams

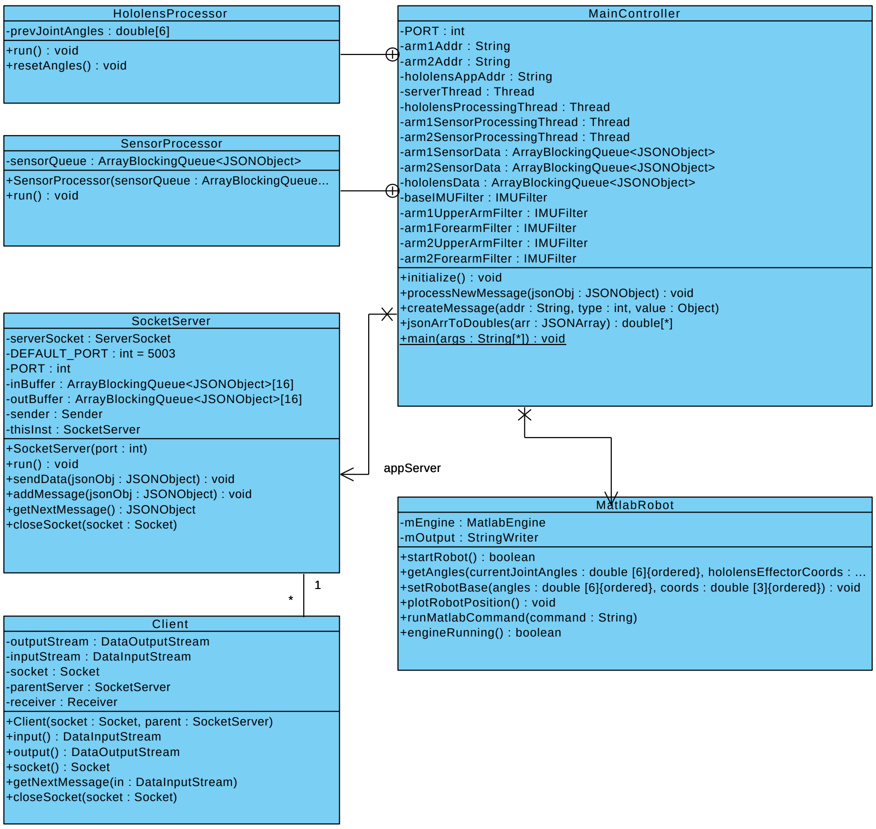


Figure A.1: MainController and related classes

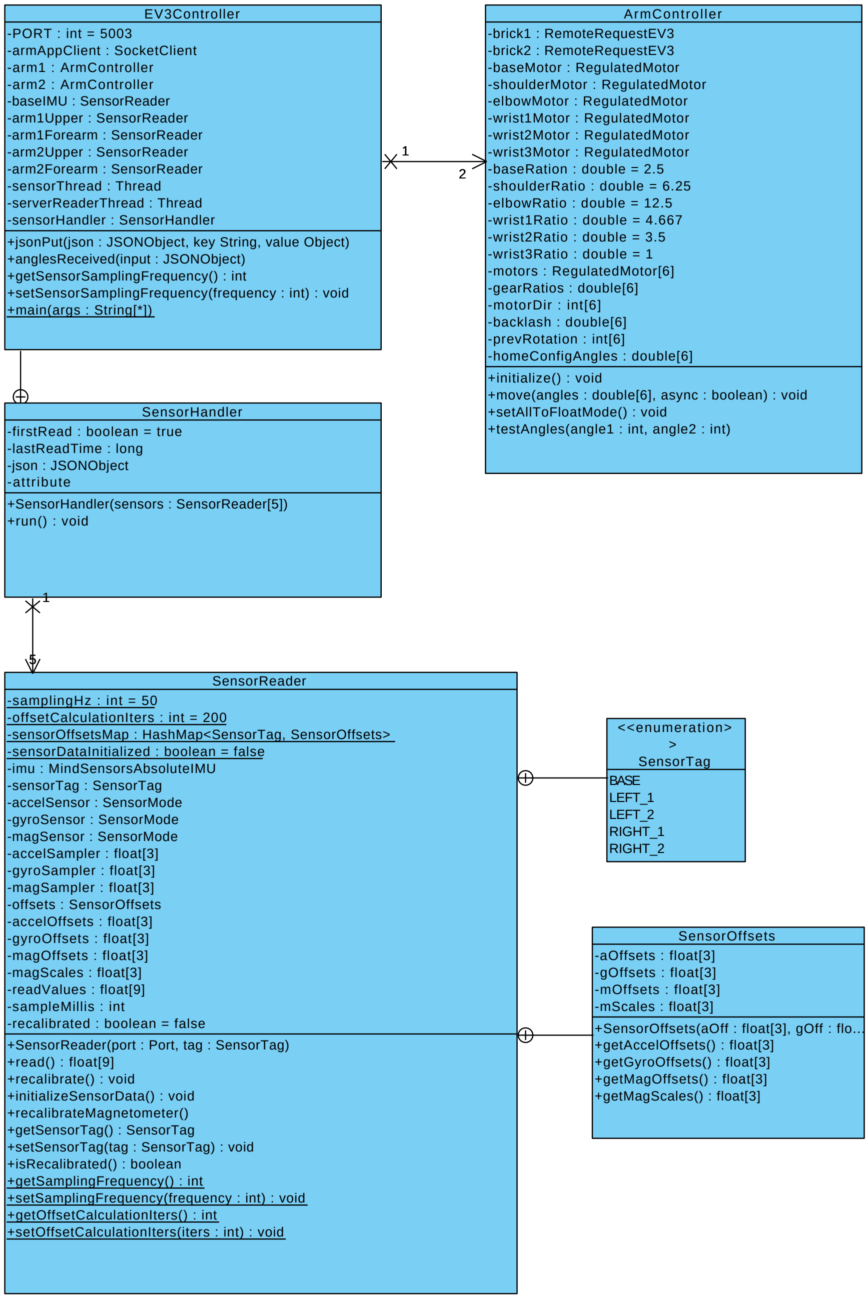


Figure A.2: EV3Controller and related classes

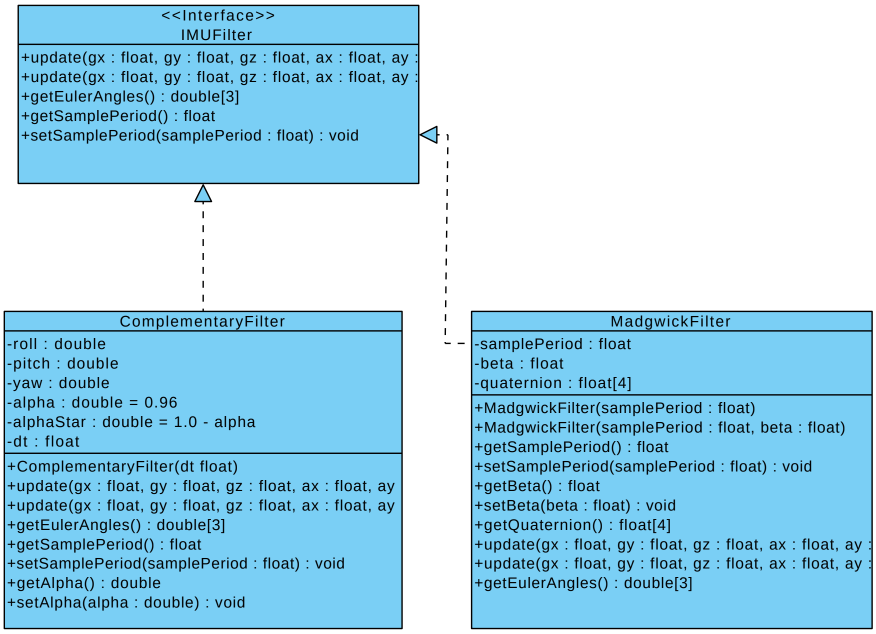


Figure A.3: IMUFilter classes

Bibliography

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

[2] Andreas Aristidou and Joan Lasenby. FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem. *Graphical Models*, 73(5):243–260, 2011.

[3] S. James Biggs and Mandayam A. Srinivasan. Haptic Interfaces. In *Handbook of Virtual Environments*, pages 93–116. Lawrence Erlbaum Assoc Inc, February 2002.

[4] Mourat Bouzit, Grigore Burdea, George Popescu, and Rares Boian. The Rutgers Master II-new Design Force-Feedback Glove. *IEEE/ASME Transactions on Mechatronics*, 7(2):256–263, June 2002.

[5] Jurgen Broeren, Martin Rydmark, and Katharina Stibrant Sunnerhagen. Virtual Reality and Haptics as a Training Device for Movement Rehabilitation After Stroke: A Single-Case Study. *Archives of Physical Medicine and Rehabilitation*, 85(8):1247–1250, August 2004.

[6] Grigore C. Burdea and Philippe Coiffet. *Virtual Reality Technology.* Wiley-IEEE Press, 2 edition, July 2003.

[7] Tom Chandler, Maxime Cordeil, Tobais Czauderna, Tim Dwyer, Jaroslaw Glowacki, Catagay Goncu, Matthias Klapperstueck, Karsten Klein, Kim Marriott, Falk Schreiber, and Elliot Wilson. Immersive Analytics. In *Proceedings of BDVA 2015, IEEE International Symposium on Big Data Visual Analytics*, pages 73–80, Hobart, Australia, September 2015.

[8] Shane Colton. The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform. Technical report, Massachusetts Institute of Technology, Cambridge, USA, June 2007.

[9] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB*. Springer, 2 edition, May 2017.

[10] National Research Council. *Virtual Reality: Scientific and Technological Challenges*. The National Academies Press, January 1995.

[11] Judith E. Deutsch, Alma S. Merians, Grigore C. Burdea, Rares Boian, Sergei V. Adamovich, and Howard Poizner. Haptics and Virtual Reality Used to Increase Strength and Improve Function in Chronic Individuals Post-Stroke: Two Case Reports. *Neurology Report*, 26(2):79–86, January 2002.

[12] Sean Weston Follmer. *Dynamic Physical Affordances for Shape-changing and Deformable User Interfaces*. PhD thesis, Massachusetts Institute of Technology, June 2015.

[13] James J. Gibson. Observations On Active Touch. *Psychological Review*, 69(6):477–490, November 1962.

[14] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. Haptic Interfaces and Devices. *Sensor Review*, 24(1):16–29, 2004.

[15] David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. Vergence-Accommodation Conflicts Hinder Visual Performance and Cause Visual Fatigue. *Journal of Vision*, 33(8):1–30, March 2008.

[16] Trevor Hogan and Hornecker Eva. How Does Representation Modality Affect User-Experience of Data Artifacts? In *Proceedings of HAID 2012, International Conference on Haptic and Audio Interaction Design*, pages 141–151, Lund, Sweden, August 2012.

[17] Carmen Hull and Wesley Willett. Building with Data: Architectural Models as Inspiration for Data Physicalization. In *Proceedings of CHI 2017, ACM Conference on Human Factors in Computing Systems*, pages 1217–1228, Denver, USA, May 2017.

[18] 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.

[19] 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 CHI 2015 Extended Abstracts, ACM Conference on Human Factors in Computing Systems*, pages 687–694, Seoul, Republic of Korea, April 2015.

[20] Hiroshi Ishii and Brygg Ullmer. Tangible Bits: Towards Seamless Interfaces Between People, Bits and Atoms. In *Proceedings of CHI 1997, ACM Conference on Human Factors in Computing Systems*, pages 234–241, Atlanta, USA, March 1997.

[21] Yvonne Jansen. *Physical and Tangible Information Visualization*. PhD thesis, Université Paris-Sud - Paris XI, March 2014.

[22] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. Evaluating the Efficiency of Physical Visualizations. In *Proceedings of CHI 2013, ACM Conference on Human Factors in Computing Systems*, pages 2593–2602, Paris, France, April 2013.

[23] 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 CHI 2015, ACM Conference on Human Factors in Computing Systems*, pages 3227–3236, Seoul, Republic of Korea, April 2015.

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

[25] Katherine Kuchenbecker, David Ferguson, Michael Kutzer, Matthew Moses, and Allison M. Okamura. The Touch Thimble: Providing Fingertip Contact Feedback During Point-Force Haptic Interaction. In *Proceedings of Haptics 2008, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 239–246, Reno, USA, March 2008.

[26] Alastair Lansley, Peter Vamplew, Philip Smith, and Cameron Foale. Caliko: An Inverse Kinematics Software Library Implementation of the FABRIK Algorithm. *Journal of Open Research Software*, 36(4), 2016.

[27] Susan J. Lederman and Roberta L. Klatzky. Hand Movements: A Window Into Haptic Object Recognition. *Cognitive Psychology*, 19:342–368, 1987.

[28] Susan J. Lederman and Roberta L. Klatzky. Haptic Perception: A Tutorial. *Attention, Perception, & Psychophysics*, 71(7):1439–1459, October 2009.

[29] Jinha Lee, Rehmi Post, and Hiroshi Ishii. ZeroN: Mid-air Tangible Interaction Enabled by Computer Controlled Magnetic Levitation. In *Proceedings of UIST 2011, Annual ACM Symposium on User Interface Software and Technology*, pages 327–336, Santa Barbara, USA, October 2011.

[30] Daniel Leithinger, Sean Follmer, Alex Olwal, and Hiroshi Ishii. Shape Displays: Spatial Interaction with Dynamic Physical Form. *IEEE Computer Graphics and Applications*, 35(5):5–11, September 2015.

[31] Thomas H. Massie and J. Kenneth Salisbury. The PHANTOM Haptic Interface: A Device for Probing Virtual Objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, USA, November 1994.

[32] W. A. McNeely. Robotic Graphics: A New Approach to Force Feedback for Virtual Reality. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 336–341, Seattle, USA, September 1993.

[33] Dan Morris, Hong Tan, Federico Barbagli, Timothy Chang, and Kenneth Salisbury. Haptic Feedback Enhances Force Skill Learning. In *Proceedings of WHC 2007, EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 21–26, Tsukuba, Japan, March 2007.

[34] Ken Nakagaki, Artem Dementyev, Sean Follmer, Joseph A. Paradiso, and Hiroshi Ishii. ChainFORM: A Linear Integrated Modular Hardware System for Shape Changing Interfaces. In *Proceedings of UIST 2016, Annual Symposium on User Interface Software and Technology*, pages 87–96, Tokyo, Japan, October 2016.

[35] Jakob Nielsen. *Usability Engineering*. Academic Press Inc, July 1993.

[36] Michal Nowicki, Jan Wietrzykowski, and Piotr Skrzypczynski. Simplicity or Flexibility? Complementary Filter vs. EKF for Orientation Estimation on Mobile Devices. In *Proceedings of CYBCONF 2015, IEEE International Conference on Cybernetics*, pages 166–171, Gdynia, Poland, June 2015.

[37] Sebastian O.H. Madgwick. An Efficient Orientation Filter for Inertial and Inertial/Magnetic Sensor Arrays. 2017.

[38] Claire O'Malley and Danae Stanton Fraser. *Literature Review in Learning with Tangible Technologies*. Futurelab, January 2005.

[39] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24:45–77, 2008.

[40] K. Salisbury, F. Conti, and F. Barbagli. Haptic Rendering: Introductory Concepts. *IEEE Computer Graphics and Applications*, 24(2):24–32, March 2004.

[41] Sue Ann Seah, Bruce W. Drinkwater, Tom Carter, Rob Malkin, and Sriram Subramanian. Dexterous Ultrasonic Levitation of Millimeter-Sized Objects in Air. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 61(7):1233–1236, July 2014.

[42] Beat Signer and Timothy J. Curtin. Tangible Holograms: Towards Mobile Physical Augmentation of Virtual Objects. Technical report, wise-2017-01, WISE Research Lab, 1050 Brussels, Belgium, March 2017.

[43] Simon Stusak, Moritz Hobe, and Andreas Butz. If Your Mind Can Grasp It, Your Hands Will Help. In *Proceedings of TEI 2016, International Conference on Tangible, Embedded and Embodied Interaction*, pages 92–99, Eindhoven, The Netherlands, February 2016.

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

[45] Brygg Ullmer and Hiroshi Ishii. Emerging Frameworks for Tangible User Interfaces. In *Human Computer Interaction in the New Millenium*, pages 579–601. Addison-Wesley Longman, August 2001.

[46] Brygg Ullmer, Hiroshi Ishii, and Robert J.K. Jacob. Tangible Query Interfaces: Physically Constrained Tokens for Manipulating Database Queries. In *Proceedings of Interact 2003*, pages 279–286, Zurich, Switzerland, September 2003.

[47] Mikael Wiberg. Interaction, New Materials & Computing: Beyond the Disappearing Computer, Towards Material Interactions. *Materials and Design*, 90:1200–1206, May 2016.

[48] Yuanxin Wu, Danping Zou, Peilin Liu, and Wenxian Yu. Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering. *IEEE Transactions on Control Systems Technology*, PP(99):1–8, 2017.

[49] Jack Zhao and Andrew Vande Moere. Embodiment in Data Sculpture: A Model of the Physical Visualization of Information. In *Proceedings of DIMEA 2008, International Conference on Digital Interactive Media in Entertainment and Arts*, pages 343–350, Athens, Greece, September 2008.