Enhancing Touchless Interaction with the Leap Motion using a Haptic Glove

Viet T. Nguyen

Master's thesis

ITÄ-SUOMEN YLIOPISTO

School of Computing

Computer Science

May 2014

UNIVERSITY OF EASTERN FINLAND, Faculty of Science and Forestry, Joensuu

School of Computing

Computer Science

Viet T. Nguyen: Enhancing Touchless Interaction with the Leap Motion using a Haptic Glove

Master's Thesis, 67 p., 2 appendices (2 p.)

Supervisors of the Master's Thesis: Susanne Mäkelä

May 2014

Abstract: Touchless interaction provides users new ways to interact with computers by using their bodily gestures. The touchless nature of this type of interaction also presents a couple of problems which are the lack of haptic feedback and unintended inputs. Motivated to mitigate the shortcomings of touchless interaction, the objective of this research was to study the feasibility of using haptic feedback in touchless interaction. In order to accomplish the objective, the areas of touchless interaction and haptic feedback were studied and a framework was developed to guide the design of touchless interfaces using haptic feedback. This framework was used to develop a prototype system that was used to evaluate the feasibility of using a haptic glove for providing feedback in touchless interaction.

The results of the prototype evaluation phase showed that the use of haptic feedback in touchless interaction is feasible when a number of conditions are met. The main bottleneck of user performance in the evaluation of this research's prototype was the erratic tracking device and the delay of the haptic glove. Therefore, the most important condition is that the device tracking user gestures should perform optimally and the device providing feedback should be responsive. The frustration and attention level of the user, the understandability and learnability of the interface's gesture vocabulary, and the association of haptic feedback to the gesture vocabulary are the remaining conditions that need to be considered to enable the user of haptic feedback in touchless interaction.

Keywords: touchless interaction, gestural interaction, natural user interface, haptic feedback, somesthesis, Leap Motion, haptic glove

Foreword

I want to extend my gratitude to my supervisor, Ms Susanne Mäkelä, who has helped me carry out this research from the beginning to the end, and to Mr Harri Karhu without whom the Leap Motion Haptic Glove would never have been developed.

I am also very grateful for the support of my loving family and friends throughout this research.

List of abbreviations

2D: 2 dimensional

3D: 3 dimensional

API: Application programming interface

GUI: Graphical user interface

GV: Gesture vocabulary

HCI: Human computer interface

Leap: Leap Motion controller

LMHG: Leap Motion Haptic Glove

Mic: microphone

NUI: Natural user interface

OS: Operating System

PC: Personal computer

VR: Virtual reality

WIMP: Window Icon Menu Pointer

Contents

1 Introduction

The emergence of touchless interaction in the last few decades has enabled new interaction paradigms that extend traditional input mechanisms such as using keyboard and mouse. Touchless interaction devices such as the Microsoft Kinect have formed part of a broader spectrum of innovations that is characterized as natural user interfaces (O'Hara, Harper, Mentis, Sellen & Taylor 2012). These touchless interaction devices give opportunities for exploring new ways of interacting with the digital world without touch, and provide opportunities for humans to manipulate digital objects as though they were real-world objects. Thus, it is important to realize the characteristics and challenges of touchless interaction in order to design experiences which can enhance human-computer interaction.

Touchless interaction is a type of interaction that can take place without mechanical contact between the human and any part of the artificial system (De-La-Barre, Chojecki, Leiner, Muehlbach & Rushchin 2009, 161). For example, interacting with a television from a distance by means of its remote controller is not touchless, while interacting with a Microsoft Xbox via a Kinect controller is. Touchless interaction involves bodily gestures and movements (O'Hara et al. 2012, 2). In this thesis, the method of touchless interaction that is studied is hand-based gestures and movements which are interpreted using the Leap Motion controller (henceforth Leap).

Natural user interfaces (henceforth NUIs) are user interfaces which are natural in the way users interact and feel when using them (Wigdor & Wixon 2011, 9). According to Wigdor and Wixon (2011), the term *natural* in NUI is used not to describe the user interface itself but a property that is external to it. O'Hara et al. (2012, 6) argues that naturalness lies neither in the physical form of a technology nor any interface that is described as natural, but in how the physical form and interface adheres to the practices of the communities that uses them. Moreover, NUI is not about which technologies guarantee a more natural user interface, but about leveraging these technologies "to better mirror human capabilities, optimize the path to expert, apply to given contexts and tasks, and fulfill users' needs" (Wigdor & Wixon 2011, 9).

A number of recent studies on touchless interaction and the associated natural user interfaces have given implications on how these technologies could work, and also presented applications of these technologies (e.g. Fikkert, Vet & Nijholt 2010; O'Hara, Gonzalez, Sellen, Penney, Varnanas, Mentis, Criminisi, Corish, Rouncefield, Dastur & Carrell 2014; Placitelli & Gallo

2014; Wachs, Kölsch, Stern & Edan 2011; and Spano 2012). Despite discussing the challenges of touchless interaction and related technologies, these researches do not cover, or cover very little, the use of haptic technologies (haptics) in touchless interaction.

A considerable challenge of touchless interaction is the inability to distinguish between intended and unintended gestures and movements. Spano (2012, 433) refers to this challenge as the well-known Midas Touch problem. Another similar term to the Midas Touch problem is the "live mic" which describes this "always-on nature of in-air gesturing", explained in Wigdor and Wixon's (2011, 98) work. Another challenge to touchless interaction is that Norman (2010) has pointed out that gestures do not leave behind record of their path and provide either no response or wrong responses, therefore lacking critical clues for successful human interaction. Other challenges of touchless interaction include difficulty to remember gestures (Norman 2010), lack of precise tuning and refining of manipulations (O'Hara 2012, 8), and choosing an understandable gesture vocabulary (Spano 2012, 433). To overcome some of these challenges, Norman (2010) suggests adding conventional interface elements such as menus and help systems, as well as incorporating feedback and visual guides.

In this thesis I will explore the use of haptics as well as novel hand-based gestures and natural interface elements to enhance touchless interaction. The use of haptic feedback in touchless interaction is not a solution for the Midas Touch problem, as for this problem there are other solutions which is discussed in sub-chapter 2.3. For the other challenges of touchless interaction as described above, I will create a glove that provides haptic feedback. The next section describes more in depth the motivation of this thesis and provides a foundation for formulating research objectives and research questions.

1.1 Motivation

Touchless interaction has contrasting properties compared to touch-based interactions, and one of the fundamental properties which touchless interaction inherently lacks is haptic feedback (O'Hara et al 2012). In addition, whereas touch-based interaction enables visual, auditory, and haptic feedback (such as computer screen, speakers, and mouse input), touchless interaction uses only visual and auditory senses, leaving haptic feedback out of the user's interaction experience. Therefore, the motivation of this thesis lies in the exploration of the use of haptics to overcome the challenges posed in touchless interaction.

Haptics, or more precisely *somesthesis*, is critical for normal human functioning in many levels, from controlling the body to perceiving, learning, and interacting with the environment (Robbles-De-La-Torre 2006). Although Robbles-De-La-Torre (2006) emphasizes on the importance of haptics in real or virtual environments (VEs), the argument raised from his work can be applied to the type of touchless interaction addressed in this thesis. Chapter 3 discusses more in detail on the definitions of the terms haptics and somesthesis.

1.2 Objective

The objective of this research is to evaluate the potential of a haptic glove to enhance touchless interaction. This objective is the result of presenting the challenges of touchless interaction and the potential benefits of haptics in mitigating those challenges. To achieve this objective, this research first studies the concepts of touchless interaction, natural user interfaces, and haptics along with related concepts and areas. A few examples of recent related applications are also given. Secondly, this research presents a framework for designing and implementing a natural user interface using haptics-enabled touchless interaction. Thirdly, the framework is used to implement a working prototype of touchless interaction with a haptic glove. Lastly, the prototype is evaluated based on user testing. The results of this research will inform the design and implementation of natural user interfaces based on touchless interaction and haptic feedback.

1.3 Research questions

Four research questions were defined in order to achieve the objective of this research.

1. What are the concepts and technologies behind touchless interaction and haptics?

The concepts and technologies behind touchless interaction and haptics must be studied before understanding how haptics can enhance the experience of touchless interaction. The definitions of touchless interaction and haptics are laid down along with analyses of the recent applications in these fields. The most fundamental concepts and technologies behind touchless technologies and haptics are studied in detail in order to provide a foundation to design and build the prototype application.

2. What are the benefits as well as challenges of using touchless interaction, and what is the feasibility of applying haptics to enhance touchless interaction?

After understanding the technologies and concepts behind touchless interaction and haptics, these two fields are analyzed, addressing the advantages as well as challenges of these fields. A strong emphasis is placed on the challenges of touchless interaction, in order to shed light on how haptics can be used to address them. By analyzing existing applications that combine haptics with touchless interaction, this research will build a solid framework on how these two fields can be effectively combined to enhance the interaction.

3. How can a haptic glove be used to enhance touchless interaction?

After analyzing touchless interaction and haptics, a touchless interaction using haptics framework is defined and elaborated. Subsequently, a functional prototype in the form of a 3 dimensional (3D) file browser application that uses a haptic glove demonstrating the feasibility of this proposed framework is explained and implemented. In order to do so, a design of the user interface of the application is described in accordance with the framework. In addition, the setup and configuration of the hardware technologies that is used in creating the prototype are laid out. The underlying software system connecting the hardware and user interface is subsequently explained.

4. Can providing haptic feedback to touchless interaction enhance the user performance and user experience of a touchless interface?

This question is meant as a guide to find out whether or not the integration of haptics will mitigate the challenges of touchless interaction and in turn enhance the user experience with touchless interaction. After answering to the previous research questions, a user testing phase will begin in which users evaluate their experiences when using the prototype application. The test will present to users two use cases, one of touchless interaction with the haptic glove, and the other without the haptic glove. The test results will in turn provide an answer to this research question.

1.4 Research methodology

The chief research methodology that is applied in this thesis is exploratory research. First, primary and secondary literature sources are reviewed and analyzed in order to find answers to the research questions above. After sufficient information has been studied to answer the research questions, a framework is then constructed. This framework would then serve as a blueprint to implement a functional prototype of an application using touchless interaction and haptic technologies. Lastly, the prototype is tested amongst students, researchers, and staff members of the University of Eastern Finland, who will act as end users to the prototype.

According to Joppe (2014), *exploratory research* is commonly used when a problem or scope has not been clearly defined as yet. Exploratory research is relevant for this research due to the novelty of the thesis topic and insights that this research will bring upon reaching its objective. Touchless interaction has been studied for decades (O'Hara et al. 2012, 2), but designing touchless interaction that meets a quality of being a *natural* interaction still requires much research. Also, based on some of the work mentioned in the beginning of this chapter, the use of haptics in touchless interactions does not yet have firm groundings on which to draw any solid conclusions. Furthermore, by reviewing literature concerning touchless interaction and natural user interface, critical insight into designing such interface systems is obtained. This research also determines whether or not the type of feedback being explored would improve touchless interaction.

In the evaluation phase of this research I will collect data with questionnaires from participants and record the tests with a video camera. The purpose of the evaluation phase is to evaluate the use of a haptic glove providing feedback to a prototype file browser application that is controlled using touchless interaction. The type of haptic feedback under evaluation is vibrotactile feedback which is explained in more detail in section 3.1. The evaluation phase also helps refine the prototype application in that through a series of iterations during which software bugs are fixed and changes are made.

To analyze the video recordings, I will count the gesture types that the participants made during the testing sessions and based on which determine their performance level. I will also make note of the user's behavior and their overall experience. The questionnaires are analyzed to gain an understanding of the user's feedback on their experience with using the haptic glove as well as their experience with using touchless interaction in general. Their recorded performance level combined with their feedback give an indication of the feasibility of the use of haptic feedback in touchless interaction.

1.5 Structure of thesis

The following chapters 2 and 3 present the areas of natural user interface, touchless interaction, and haptics research and their associated technologies. Specifically, chapter 2 analyzes in detail the strengths and challenges of touchless interaction. Thus, chapters 2 and 3 answer the first and second research questions. An introduction to the Leap and the design process of the Leap Motion Haptic Glove (LMHG) is presented in chapter 4. Chapter 5 draws experiences and insights from the previous chapters to construct a framework for designing a natural user interface using touchless interaction enhanced with haptic feedback. Subsequently, a working prototype of a natural user interface system that combines the Leap and the LMHG based on the framework is constructed. The last section of chapter 5 discusses the integration of the LMHG system. Chapters 4 and 5 answer the third research question. Chapter 6 describes user groups, user testing scenarios, and method for data collection and analysis. Chapter 7 discusses the results of the evaluation phase of this research and in turn answers the last research question. Chapter 8 draws conclusions for this research based on the results gathered from the evaluation phase as well as insights attained by studying the background area.

2 Natural user interfaces and touchless interaction

This chapter covers NUIs and touchless interaction, their background and underlying concepts, and their applications. Although some sources may refer to touchless interaction as NUIs (O'Hara 2012, 2), NUIs as gestural interfaces (Malizia & Bellucci 2012, 36) (gestural interfaces is also another term for touchless interaction, to be discussed in sub-chapter 2.2) and vice versa, this chapter aims to establish the definitions of each of these terms, and their relationship to one another.

2.1 Natural user interfaces

The introduction chapter gave a definition of NUI that relies on the understanding of the term *natural.* The consensus drawn out by Wigdor and Wixon (2011) and O'Hara et al. (2012) is that the naturalness of an interface lies not in an interface itself, but how such interface behaves in its context of use. Malizia and Bellucci (2012) on the topic of gestural interfaces describe NUIs as interfaces which are in contrast with the traditional computer interfaces having artificial control devices that had to be learned, such as the mouse and keyboard. Similar to the consensus drawn by the authors mentioned above, Malizia and Bellucci believe that when using a natural interface, users should be able to use the same gestures that they use to interact with objects in everyday life. But because we have grown up and have been nurtured in varying environments, these everyday gestures may vary. Câmara (2011) of YDreams, an international leader in interactivity, gives a concise definition of NUIs which are touch and motion based interfaces that are replacing the traditional mouse and keyboard. These definitions on NUIs serve as a theoretical foundation to this thesis, and ultimately to the design and implementation of the prototype application which is one of the outcomes of this thesis.

NUI is considered to be the next potential phase in the evolution of computing. It promises to further reduce the barriers of computers and enhancing the powers of the user. However, NUI should not be seen as a replacement to GUI in the future, nor should it be judged based on single instances of technologies which explore its concepts. The success or failure of technologies which attempts to implement NUI should not be seen as a predictor of NUI's future. However, a safe prediction is that NUI will not disappear, and will either be seen in a niche area, or dominate the whole computing area. (Wigdor & Wixon 2011.)

To aid the design of the prototype application of this thesis, Wigdor and Wixon's framework to create natural user interfaces is used as a reference. The figure below presents this framework.

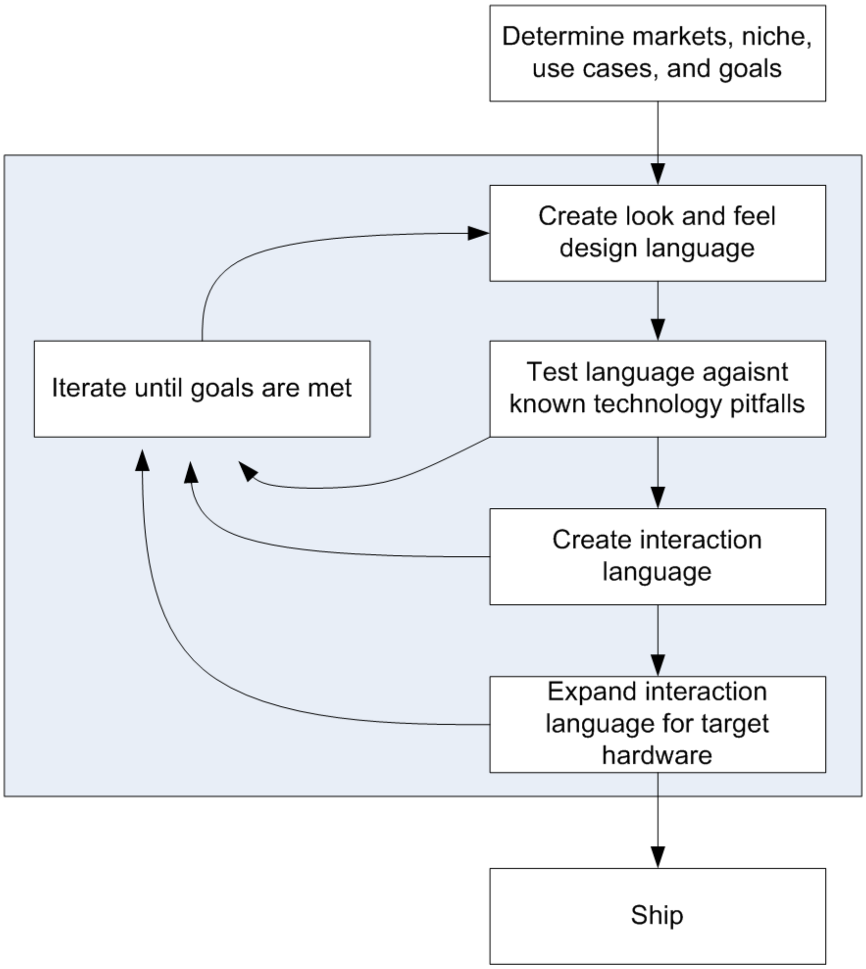

Figure 1. Wigdor and Wixon's (2011, 7) framework for the creation of NUIs

Figure 1 shows that the initial steps to designing a NUI involves determining markets, niche, use cases, and goals followed by creating the look and feel design language (step 2). The subsequent steps are performed iteratively until goals are met. Iterated steps restart from step 2.

Touchless interaction is one of many modalities which can be used to create NUIs, amongst other modalities such as touchscreen devices and voice commands. They offer the opportunity to create NUIs, but by themselves they do not guarantee naturalness (Wigdor & Wixon 2011). Thus, this is the difference between the two terms touchless interaction and NUI, and in my opinion why the terms NUI and touchless interaction should not be used interchangeably and instead should be understood in their separate meanings. Touchless interaction may not necessarily signify a NUI, and a NUI does not necessarily mean it uses touchless interaction. The next section explores the concepts behind touchless interaction, followed by its characteristics, benefits, drawbacks, and application.

2.2 Touchless interaction

As stated in the introduction chapter, touchless interaction is interacting with user interfaces without physical contact. Touchless interaction is also referred to as in-air gesturing, (Wigdor and Wixon 2011, 97), hand-gestures (Wachs et al. 2011, 60), and body-based interface (O'Hara et al. 2012). These terms are used interchangeably in this thesis when referencing the works of authors who use them. Nonetheless, to keep the text of this thesis consistent, touchless interaction is the prominent term that is used. Although touchless interaction can involve the use of all parts of the body, this thesis is concerned only with the position changes and gestures of the hands. As this thesis explores the use of hand gestures to manipulate user interfaces, an understanding of what is a gesture, and other types of hand movements, needs to be attained.

Karem and Schraefel (2005) identified different forms of gestural actions based on their review of 40 years of literature on gesture-based actions: *deictic* gestures for pointing, *manipulative* gestures for controlling objects, *semaphoric* gestures for symbolizing an object or action as a way to communicate, *language* gestures such as sign language, and *gesticulation* which are gestures that accompany speech. The types of gestures concerned in this research to create the interaction language for the prototype application are mainly deictic and manipulative gestures with a couple of semaphoric gestures.

The properties of touchless interaction can be understood by referring to contrast points of touch-based interaction. Although describing the list of properties based on this method will not cover all of the properties of touchless interaction, it will help in comparing touchless interaction to the different communities of practice and setting. The following table and the subsequent descriptions of its contents are cited from O'Hara et al. (2012, 7-8). Table 1 below lists the properties of touchless interaction with regards to their touch-based interaction contrast.

Table 1. Contrasting characteristics of touch vs. touchless interaction

Touch	Touchless
co-proximate with surface	distant from surface
transfer of matter	no transfer of matter
pressure on surface	no pressure on surface
momentum of object	no momentum
attrition and wear of surface	no attrition or wear
movement constrained by surface	freedom of movement
haptic feedback	no haptic feedback

The first property, distant from surface, is the basis for touchless interaction's definition which is explained above. Where touch-based interaction requires users to be co-proximate with the surface they are touching, touchless interaction allows users to interact from a distance. The range of the proximity between the user and the surface of the system depends on the sensing technology.

Due to transfer of matter, no pressure on surface, and no momentum, touchless interaction systems have no attrition or wear. The property of freedom of movement is due to the fact that movement of the user when using touchless interaction technologies is no longer restricted to the confines of the technology's surfaces.

The final property is the lack of haptic feedback. With touch-based interaction, contact with the technology's surface provides haptic feedback through which manipulations can be tuned and refined on a moment-to-moment basis. In touchless interaction, the source of haptic feedback is absent, and which strips the user's ability to finely tune and refine manipulations in the moment of interaction.

There are several benefits of touchless interaction identified through studies by Wachs et al. (2011) and De-La-Barre et al. (2009). First, touchless interaction allows maintaining total sterility which is useful in health-care environment where sterility is a top priority. Second, touchless interaction overcomes physical handicaps in that it provides means for people with physical handicap and elderly persons to control devices and appliances. For instance, wheelchairs that are enhanced with robotics and intelligence are able to recognize hand and face gesture commands (Kuno, Murashima, Shimada & Shirai 2000). Also, people with physical handicaps can control robots with gestures when other methods of interaction are limited or impossible without specialized keyboards or robot controls. Third, the exploration and manipulation of large complex data volumes may see benefit from 3D interaction rather than the traditional 2D interaction. Fourth, touchless interaction is useful in vandalism-prone locations in that the devices used for touchless interaction such as the display device and the sensor can be kept behind secure structures such as glass walls. Fifth, touchless interaction enables the co-located joint usage of interactive systems, as such in classroom environments using large displays. Sixth, the 3D interaction capability of touchless interaction as denoted in the third benefit point can also be used to move a real or virtual object in all three dimensions. Last, touchless interaction is useful in environments requiring users to comprehending a system's input devices as quick as possible. These environments may include doors and lights in public places, conveyor transport system, and service robots. The first, fourth, and fifth benefits are derived from the property that touchless interaction does not come into direct contact with the display and/or input device. The second, third, and sixth benefits arose from the freedom of movement property of touchless interaction.

2.3 Drawbacks of touchless interaction

Evaluating the use of haptics in touchless interaction is the objective of this thesis, and the motivation of this objective stems from the drawbacks of touchless interaction. In this section I will analyze the drawbacks of touchless interaction in order to understand and to provide solutions to these drawbacks. After explaining each drawback, potential solutions to that drawback are provided. The explanation to the use of haptics as a solution to mitigate these drawbacks is elaborated in sub-chapter 4.4.

The first drawback of touchless interaction is the inherent *live mic* (Wigdor & Wixon 2011, 98) or the Midas Touch problem (Spano 2012, 433). In Wigdor and Wixon's explanation, the live mic problem refers to the always-on nature of in-air gesturing which is comparable to a microphone that is always on recording any speech or sound. This means that both intended and unintended gestures are recorded by the touchless interface. Moreover, Norman (2010) points out that because gestures are short-lived they do not leave behind any record of their path, it is not possible to determine whether or not the gesture gives the right or wrong response and how to differentiate between them. As can be seen, possible solutions to the live mic problem are to provide ways to differentiate intended inputs from unintended ones, and to provide feedback to these inputs.

A solution to the live mic problem is to provide a "clutch" (Wigdor & Wixon 2011, 98-101; see also O'Hara et al. 2014, 74-76). A clutch in a gestural sense is an indicator to start or stop the recording of a gesture. An example of a clutch in touch-based input is the act of touching the finger across a touchscreen device to start a gesture, and lifting the finger to end it. Another solution is to provide multi-modal input. For example, to combine hardware buttons with software buttons (such as holding down a keyboard key, and using the mouse to click on a software button), or to combine voice input with in-air gestures (such as saying a word to a touchless interface to begin recognizing hand gestures). Although not a solution in itself, haptic feedback may act as an assistive technology or be used along with the two solutions mentioned above. Haptics feedback is relevant form of feedback when used in the solutions to the live mic problem because it makes use of the user's sense of touch, a sense which the user naturally uses when they are interacting with their hands. Auditory feedback coupled with visual elements mentioned above is also relevant in solving this problem.

The second challenge of touchless interaction is the lack of haptic feedback, as briefly denoted by O'Hara et al. (2012, 8). As discussed above on the haptic-less property of touchless interaction, manipulations made are harder to be finely tuned. This is because there is nothing for the user to hold, feel, or grasp and the movement of the user's hands has no momentum of a held object. Thus, the accuracy of the user in touchless interaction systems is entirely dependent on the agility of their limbs. The lack of haptic feedback is explained in more detail through a study of an article in sub-chapter 3.2. What haptics may offer for the users in touchless interaction systems is to simulate the presence of a virtual object with which they can feel, guide, and fine-tune their manipulations.

The interaction design challenge that both touch-based and touchless interaction face as emphasized by Norman (2013) is relevant for analysis in this thesis. Norman posed questions aiming to overcome challenges of touchless interaction which includes determining the region being sensed, the range of possible inputs, and how the product communicate the possibilities to users. Other questions were raised to tackle issues such as (1) what the users can do, (2) where and how, (3) what has happened, (4) how users get back (undo), and (5) how the product works. Issue 3 can be solved by providing feedback. Issue 5 can be solved by providing "a clear, coherent conceptual model of the operation of the product which must be provided through clues within the design itself" (Norman 2013b as cited by Norman 2013a). Issues 1, 2, and 4 can be solved through affordances and signifiers. Affordances are physical structures that enable interaction, such as the affordances of small objects enable lifting, throwing, and probing. A "signifier" is "a perceivable signal if the location and form of the possible input interaction" (Norman 2010; 2013b as cited by Norman 2013). In the case of touchless interaction, signifiers are diagrams or short animation sequences that help the user with learning the system. Haptic feedback may provide solutions to some of the issues presented by Norman which are presented in detail in sub-chapter 3.2.

2.4 Applications of touchless interaction

This section discusses in detail one application of touchless interaction, followed by brief outlines of a few other applications.

By providing a contact-free interaction, touchless interactions maintain sterility between the user and the interaction device. O'Hara et al. (2014) made a study of touchless interaction in surgery and presented a touchless system using the Microsoft Kinect that is used during image-guided vascular surgery at Guy's and St. Thomas's Hospital (GSTT) in London, UK. This touchless system is referred to as GSTT system. When using touch-based systems, surgeons cannot touch these systems without breaking asepsis. Workarounds such as having a specialized member to perform requests under the surgeon's instructions may not always achieve wanted results. Issues that arise from having specialized member include frustration and delays, the member's lack of critical clinical knowledge and interpretation, and lack of hands-on analytic and interpretive tasks that is needed by surgeons.

O'Hara et al. (2014) have met a few challenges when developing the GSTT system. One challenge that arose is the live mic problem discussed in the last section. There were confusions in recognizing false positive gestures when the surgeons are preforming gestures that accompany their speech, as well as unintended movements across the screen. O'Hara et al.'s solution was to create a clutch, or what they have described as "mechanisms to move between states of systems and engagement and disengagement, reinforced with appropriate feedback to signal the system state". Thus, they incorporated a specific gesture - the withdrawal of the arms towards the body – as well as accompanying voice commands that complement the gesture vocabulary and the changing of states.

Another challenge was to incorporate collaborative control of the system. In the case of the GSTT system, O'Hara et al. were able to provide collaboration by color-coding cursors associated with each surgeon, coupled with spoken commands to request control of the system.

O'Hara et al. (2014) concluded that their project was not only to prove the feasibility of touchless control in clinical settings but also to address design challenges such as developing the gesture vocabulary and using multi-modal inputs and specific sensing mechanisms. The contactless nature of touchless interaction can address the problems of sterility and infection control procedures in hospitals. Also, touchless interaction allows freedom in movement which enables complex 3D navigation techniques such as viewing a large and intricate volume of medical images.

There have been other gesture based applications that have facilitated the controlling of visualization displays and interacting with medical instruments. Such applications include the Face MOUSE (Nishikawa, Hosoi, Koara, Negoro, Hikita, Asano, Kakutani, Miyazaki, Sekimoto, Yasui, Miyake, Takiguchi & Monden 2003 as cited by Wachs et al. 2011) and the Gestix (Wachs et al. 2008 as cited by Wachs et al. 2011). An application using touchless technologies developed to aid patients is the Staying Alive (Becker & Pentland 1996 as cited by Wachs et al. 2011) virtual-reality-imagery-and-relaxation tool which allows cancer patients to navigate through a virtual scene using 18 Tai Chi gestures. Bartoli, Clara, Franca, and Valoriani (2013) used a range of Microsoft Kinect-enabled games to explore the effectiveness of motion-based touchless games to aid the education and development of autistic children.

In the market today, there are affordable devices enabling touchless interaction. In entertainment and especially the gaming sector, the Microsoft Kinect is a device which enables touchless interaction through bodily gestures. This device has two tracking modes: default, optimized for full-body-skeleton tracking, and seating, optimized for the tracking of the upper-torso which includes the head, shoulders, and arms. The Kinect also supports voice recognition to augment the gaming experience (O'Hara et al. 2014, 76.) The Nintendo Wii entertainment system offers the WiiMote which the player holds in his hand. Although it is not a touchless interaction in a sense that the user's hand is touching a part of the system, here the user's hand gestures are recognized when handling the WiiMote, and with the buttons present on the WiiMote, clutching operations may aid in overcoming the live mic problem (Wachs et al. 2011, 67).

The WiiMote was used in a research that created a 3D gestural interface for collaborative music creation (Bott, Crowley & Laviola Jr. 2009). In this research, the WiiMote enables 3D spatial interaction techniques to manipulate gestural interfaces that control both the timing and sound of music being played in both collaborative and single player modes. The purpose of this research is to develop a method to detect different musical gestures without explicitly selecting them. Using the WiiMote, six gestural interfaces were developed that enables the players to control one of six musical instruments by mimicking the way that each instrument is played. A multi-instrument musical interface (MIMI) was described in this research that allowed the ability to determine which instrument is being played by analyzing the nature of the hand movement.

Both the WiiMote and Kinect controllers were used in a study by Francese, Passero, and Tortora (2012) to create a 3D gestural user interaction on 3D geographical maps. The researchers created two systems for each of the controllers and evaluated and compared the subjective usability and perceived sense of presence and immersion of users when using these two systems. The conclusion of the study was that the more an interface is natural and involves the users' bodily movements, the more satisfied and more involved they are in the 3D navigation experience. The research also suggested that the classic windows, icons, menu, and pointer (WIMP) interface should be avoided whenever possible when designing gestural interfaces and instead rely on new gestures and new forms of physical commands. The reason for this avoidance of WIMP elements in a gestural interface is because gestures are not as effective to control WIMP elements as the mouse and keyboard. In games made for the Kinect and WiiMote, interactions with window and icon elements are limited to game menu control, while the main gaming experience uses gestures that are analogies to their real counterparts (Francese et al. 2012), such as swinging a tennis racket, or kicking a soccer ball.

3 Haptic technology

Before analyzing the feasibility of using haptic feedback in touchless interaction, the area of haptic technology must be understood. This chapter explains the definitions of concepts related to the area of haptic technology and discusses a range of applications that studies this area. These applications are presented to show the importance of haptic technology in a range of environments and settings so that an understanding may be reached on how it can also benefit touchless interaction.

3.1 Definitions and background

According to Robles-De-La-Torre (2010), *haptic technology*, or *haptics*, is generally referred to as the science of *touch* in real and virtual environments. This discipline thus includes not only the subfield of touch capabilities in different organisms, but also the subfield which deals with the development of systems that creates haptic virtual environments. The latter subfield is referred to as computer haptics. Moreover, Robles-De-La-Torre (2006) explains the meaning of the term *haptic* which is the active exploration of the environment usually with the hands to determine shapes and material properties.

Lederman and Klatzky (2009, 1439) describe that "the haptic system uses sensory information derived from mechanoreceptors and thermoreceptors embedded in the skin ("cutaneous" inputs) together with mechanoreceptors embedded in muscles, tendons, and joints ("kinesthetic" inputs)". Robles-De-La-Torre (2006) refers to the combination of cutaneous and kinesthetic inputs as *somesthesis*. Also, when speaking about the sense of *touch*, its meaning usually refers to cutaneous sensations. The terms *haptic* and *haptics* are increasingly used to refer to all somesthesis capabilities.

The type of haptic feedback that is explored in this thesis is *vibrotactile* feedback that is intended to simulate cutaneous sensations. According to Kaczmarek, Webster, Bach-y-Rita, and Tompkins (1991), vibrotactile feedback creates tactile sensations on the skin using mechanical vibration at frequencies of 10-500 Hz. Vibrotactile feedback is a type of haptic feedback that can be seen in education and entertainment research, amongst other areas. In education, Van Der Linden, Bird, Rogers, and Schooderwaldt (2011) studied how real-time vibrotactile feedback could assist children in learning to play the violin. Vibrotactile feedback to aid the visually impaired has been seen in studies such as that of Ghiani, Leporini, and Paterno (2009) and Shrewsbury (2011). Kim, Kim, Soh, and Yang (2006) developed a vibrotactile rendering method that was used in a mobile game to show the effectiveness and feasibility of vibrotactile feedback. Morrell and Wasilewski (2010) developed a vibrotactile seat to augment collision warning systems by displaying spatial information in the seat. A few of these studies are discussed in the next section, where the importance of haptics is explored.

3.2 Importance of haptics

This section elaborates on the importance of haptics through examples of applications of haptics in HCI.

Robles-De-La-Torre (2006) did a study on the importance of the sense of touch for humans in real and virtual environments. The term touch in this study refers to somesthesis input. In the study, the importance of touch is emphasized through analyzing cases of patients who suffer from permanent loss of the sense of touch. The patient struggles in performing relearning tasks such as walking and sitting and the ability to learn new tasks was also a problem. To cope with these struggles, the patient has to entirely rely on his sense of sight to consciously guide his limb to perform actions purposefully. Even with a fully conscious effort using his vision to aid the movement of his limbs, the patient still cannot function as precisely and speedily a normal person would.

The loss of the sense of touch for a normal person can be demonstrated by such situations as when a person sleeps on their arms, or when pressure is applied to a limb in general, or when anesthesia is administered to a part of the body. In such cases, controlling a numb hand may be difficult, for example, when grasping objects and performing skilled actions such as buttoning a shirt.

Robles-De-La-Torre then analyzed in detail the state of the absence of touch and inadequate touch information in virtual environments and HCI research. He observed that very little somesthetic feedback is provided in today’s virtual environments. He also speculates that the poor somesthetic feedback can be comparable to the loss of the sense of touch of the patient as examined above. As such, the user’s performance when using a real or virtual system may degrade when there is poor somesthetic feedback. The conclusion to Robles-De-La-Torre’s study was that somesthesis is essential for normal human functioning - from controlling the body to perceiving, interacting, and learning the environment - and that this argument strongly applies to the interfaces of real and virtual environments.

In Van Der Linden et al.'s (2011) research, the exploration of vibrotactile feedback in learning to play the violin gave insight into the deployment of haptic technology in real world uses. Vibrotactile feedback enables the violinists to be fully engaged in their learning process while still listening to notes they are producing and reading music. Visual feedback can be distracting while reading music notations, and audio feedback can interfere with the sounds of the violin. The researchers created the MusicJacket that gives haptic feedback to the violinists who would wear them when playing the violin. The MusicJacket has inertial motion sensing devices that can detect the violinist's posture and movement and then gives real-time feedback in the arm and torso area. The purpose of the MusicJacket was to improve the violinists on two areas: correct violin holding and straight bowing. The MusicJacket is outfitted with vibration motors that create the haptic feedback by creating vibrations that the violinists can feel whenever their posture is off or they incorrectly move the bow. Apart from benefits in learning to play the violin, the researchers also concluded that vibrotactile feedback can be a common vocabulary between students and teachers in movement-based learning.

Morrell and Wasilewski (2010) developed a vibrotactile car seat to improve spatial awareness while driving. They used a driving simulation environment using a PC and a haptic steering wheel and pedals to drive the virtual car. The results of this research indicated that there were small beneficial changes when vibrotactile was used when driving. A mental model of the surroundings of the car was developed by the driver through the vibrotactile feedback so that the driver can react with confidence. Although the vibrotactile seat increased the test drivers' confidence by indicating to them the location of other cars near them, it also increased the level of risk due to their sustained proximity with those cars. There were still several considerations when using haptic feedback in driving scenarios, but this research paved a way into how similar research can be carried out in order to leverage the benefits of using haptic feedback in car driving and other automotive areas.

Apart from the studies mentioned above, there have been studies on the use of haptics in various HCI fields such as virtual reality (VR) environments, touchless interaction, and mobile touch-screen, amongst others (Regenbrecht, Hauber, Schoenfelder & Maegerlein 2005; Gallotti, Raposo

& Soares 2011; Shrewsbury 2011; Truong & Yatani 2009; Cassidy, Cockton & Coventry 2010, Israr & Poupyrev 2011).

In VR environments, the research of Regenbrecht et al. (2005) argued on the limitations visual feedback in tasks requiring spatial placement of objectives relative to each other, and suggested several approaches which include force feedback, augmented reality, tactile feedback, and vibrotactile feedback. The vibrotactile feedback was chosen for Regenbrecht et al.'s research due to low-cost actuators that increase performance and reduce errors in assembly tasks. At the time of Regenbrecht et al.'s research, technologies to track hands and movement in VR environments were mechanic, acoustic, magnetic, optic, inertial, or combinations of these. The chosen tracking system was magnetic based due to its popularity in the market at that time. Optical tracking was more expensive and requires extra setup, calibration, and visible line-of-sight. From quantitative data gathered, Regenbrecht et al. determined that tactile feedback increased the overall successful feedback, and precision of control, and lowered task completion time compared to only visual feedback. Also, tactile feedback can always be perceived and not obstructed by occlusion like visual feedback. After testing using two different prototypes, the researchers concluded that the use of vibrotactile feedback in general demonstrated benefits and received positive user feedback.

Gallotti et al. (2011) developed a wireless glove (called v-Glove) that tracks the user's index finger and vibrates the tip to simulate a touch feeling. The aim of the v-Glove is to map a touch interface in a VR immersive environment to overcome the abrupt changes in switching from non-immersive modes (e.g. 2D desktop interface) to immersive modes (3D navigation and manipulation).

Shrewsbury (2011) conducted a study to help the visually impaired by enabling detection and navigation of the surrounding environment using sensory data from the depth camera system of the Microsoft Kinect coupled with haptic feedback. The resulting prototype was a haptic glove that had buzzer motors aligned to every finger except the thumb, and the individual motors would buzz according to the individual pixel of the resized depth image from a middleware used to process the Kinect's sensory data. The research showed opportunities for related research work to provide navigational assistance and object avoidance.

Israr and Poupyrev (2011) developed the Tactile Brush algorithm that creates tactile feedback using two tactile illusions - apparent tactile motion and phantom sensations. Using these tactile illusions, high-density two-dimensional tactile displays using vibrotactile arrays can be designed. Israr and Poupyrev suggested that the Tactile Brush algorithm can be flexible enough to support a wide range of applications, actuation technologies and embodiments. Despite several limitations, the Tactile Brush algorithm promises complex and rich multidimensional tactile experiences that can be used in such applications as gaming, entertainment, aids for the blind, driving and navigational aids, and mobile computing.

4 The Leap Motion Haptic Glove system

The importance of the sense of touch in touchless interaction is a topic that requires thorough research to be able to come to any solid foundations or usable knowledge. The lack of haptic feedback when using touchless interaction may pose problems, as can be seen from the last section. Also, as discussed in sub-chapter 2.3, a common solution to most of the challenges was to provide appropriate feedback. Coupled with the foundations of the importance of the sense of touch affirmed in the last section, haptic feedback is the type of feedback that should be explored as a possible solution to the challenges of touchless interaction.

This thesis will evaluate the use of haptic feedback in touchless interaction by creating a haptic feedback glove called the Leap Motion Haptic Glove (henceforth LMHG). Using the LMHG, this research will test users in two scenarios, one scenario that employs haptic feedback and another that functions without it. The result of this test will hopefully shed new light into this new topic of haptics in touchless interaction, and bring insight into future applications and frameworks.

The LMHG is a core component of the prototype, and will serve as the haptic feedback device when using the Leap with the prototype GUI. As the name implies, the LMHG relies on the Leap Motion controller to track hand position. In this chapter, an overview of the Leap and its API is laid out. The process of creating the LMHG, from initial design to fully functional prototypes is explained thereafter.

4.1 Overview of Leap Motion controller

The Leap Motion controller, the main device to be used in this research, is a type of touchless interaction device owned and manufactured by Leap Motion Inc. Originally named OcuSpec, Leap Motion Inc. was co-founded by Michael Buckwald and David Holz in 2010. After raising a seed round from venture capital firms in 2011 (Tsotsis 2011), followed by further funding from two more funding rounds (MarketWired 2012; Kosner 2013), the company launched its first product, the Leap Motion controller on 27th July 2013 (Etherington 2013).

The Leap can detect a user's hands, fingers, and finger-like objects (tools) in its inverted square pyramid field of view. The field of view has an effective range of 25 to 600 millimeters measuring from the top of the device (Leap Motion 2014). The Leap is designed so that it sits in front of the user's computer screen. Interaction is done by making gestures with the hands, fingers, or finger-like objects such as a pen or pencil. As stated in the Leap Motion Company's front page, the Leap Motion is not intended to replace a keyboard or mouse, but it is a supplementary interaction device.

Although the fingers and hands can be tracked accurately, they must be positioned on top of the Leap so that your inner palms should always be facing downwards towards the Leap. This is because the Leap will stop detecting the hands and fingers if the hands are tilted due to the fingers no longer being in its vision.

The Leap is a touchless interaction device, and it is this kind of interaction that is the motivation of this research; to see if users feel more comfortable to be able to feel haptic-feedback when they interact with the computer using the Leap.

As of November 2013, the Leap's official online application store – the Airspace – has 150 applications in its catalog (Rodriguez 2013). In the Airspace Store (2014) the applications consist of interactive games, learning applications, creative applications, music applications, and applications that aim to control the computer with the Leap (i.e BetterTouchTool, GameWave, AirInput).

4.2 Leap Motion API

According to the Leap Motion Developer website (Leap Motion 2014), the Leap can recognize three aspects of hand input. The first aspect is the ability to recognize hands, fingers, and finger-like tools (i.e. pens, pencils, paint-brushes) and provide software interfaces to get information on each of these input types. The second aspect is the recognition of gestures, such as circles, key taps, and screen taps. The last aspect is the recognition of motions of the hands, fingers, and finger-like tools such as scaling, translation, and rotation. In programming the prototype for this thesis, all of the aspects of hand input recognized by the Leap are utilized.

The prototype was created using the Unity 3D Game Development software (Unity) and scripted using C#. The Leap provides a C# API and an example project using Unity. The example project served as a basis to implement the prototype for this thesis.

To get the relevant data from the Leap to the prototype, the following programming classes were used in the Leap API (Leap Motion 2014):

- Frame: A frame object is the root of the Leap data model, and holds the data of the classes below. Each frame holds basic tracking data such as hands, fingers, and tools, and recognized gestures and other factors describing the overall motion.
- Hand, Finger: These two classes were used to track the user's hand and finger motions. Specifically, the positional and directional data of the hand that was calculated by the Leap is fed into the prototype and mapped into the 3D scene as a virtual representation of the hand.
- Gestures: Only the Swipe gesture is used in the prototype which is a long and linear movement of a hand or finger.

4.3 Previous version of the Leap Motion Haptic Glove

I developed the first version of the LMHG as a special IT project (Nguyen 2013). For this thesis work, I developed the second version of the LMHG with the help of colleague Harri Karhu, who also supported me a great deal during the making of the first version of the LMHG.

The first version of the LMHG had kinesthetic feedback, and the prototype application used was a waste sorting game that was tested with children with special needs and subsequently with a group of HCI students and staff of my department. The kinesthetic feedback was implemented by using a servo mounted on the wrist section of the LMHG that pulls strings that are tied to the end of each finger. This setup of the LMHG is to imitate the feeling of having an object in the hand when grasping it.

Overall, the outcome of the project determined that users who were tested liked the idea of having haptic feedback when using the Leap. The drawbacks observed in from this project were due to the limitations of the Leap as well as the design of the LMHG, including jittery and jumpy tracking of the hands when using the glove and even when not using the glove and lack of a firm mounting point of the servo to properly pull the fingers.

The goal of designing the second version of the LMHG with vibrotactile feedback is to simulate the sense of touch in real-time. This means that whatever the user is visually presented with in the prototype GUI, a haptic response will be given to the user immediately via the glove. Despite the fact that initially the LMHG suffered from a delay problem, the evaluation phase helped incrementally improve the LMHG and the delay problem was fixed. The following section will give an in-depth look into the technology that powers the LMHG and the development phases that the LMHG went through.

4.4 Developing the Leap Motion Haptic Glove

This research has created the LMHG that imitates cutaneous inputs using vibrotactile feedback. Specifically, cutaneous input is imitated by using vibration motors placed under the hand; under each fingers and the inside of the palm. In the first version kinesthetic feedback was planned for the glove which is imitated by using a servo attached to a mount point on the glove near the wrist section that pulls the fingers back using attached fishing lines. However, due to the complexity involved in adding both cutaneous and kinesthetic feedback into a single glove, as well as the fact that I had already tested the previous version of the haptic glove with kinesthetic feedback, cutaneous feedback was chosen to be the sole feedback modality for the second version of the LMHG.

The LMHG was created using a right-handed cotton glove which serves as a frame to hold the vibration motors and the Arduino board in place. With the help of Harri Karhu, my colleague, the LMHG was assembled using 16 vibration motors on the underside of the glove, and was connected to a shift register array board which in turn was connected to the Arduino Mini Pro board. A photo of the underside of the glove is shown below.

Figure 2. The underside of the Leap Motion Haptic Glove with 16 vibration motors

Figure 2 illustrates that the 16 vibration motors are all placed on the underside of a right-handed cotton glove. The wirings of the motors meet in the central Arduino board that is on the other side of the glove. Specifically, the 16 vibration motors are connected sequentially to a shift register. A shift register is a type of sequential logic circuit that turns a serial input to a parallel output. A shift register works by shifting one bit at a time from the input pin through each of the bits of the shift register until it comes out the output pin. A clock signal determines when each shift happens. There is also another clock signal called a latch that can transfer each bit in the shift register into a storage register where the bits are outputted. A pin called the enable pin controls the output of the bits that are in the storage register, without affecting the bits in the shift register. (Educypedia 2013; Szczys 2011.)

Mr Karhu designed a 16-bit shift register, thus the bit that travels into the input pin goes through each of the 16 bit positions in the shift register. The program that controls the motors acts in a way that allows a single 1 bit travelling through the shift register at a single time. Thus only one position of the 16 bit positions has a value of 1 after every shift, and the other positions have a value of 0. Every time the clock signal is called to shift the bit, a latch signal will also be called, copying the values of the shift register into the storage register. When the enable pin is set to high (value of 1), the motor that is attached to the position in the storage register that currently has a value of one will turn on. Figure 3 below describes the working of the shift register used in the LMHG. Notice that the figure shows an 8 bit shift register for demonstration purposes.

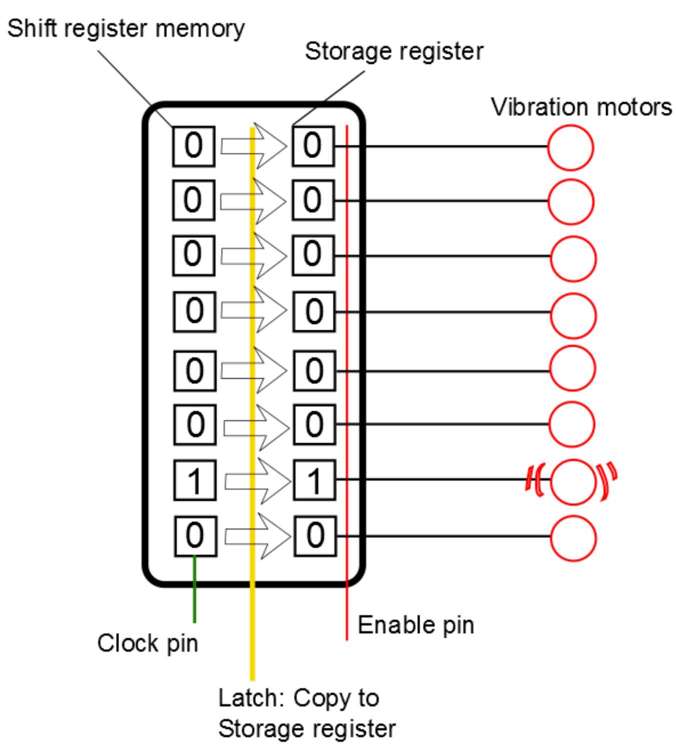

Figure 3. Function of a shift register

The reason for the use of shift registers was to be able to support a large number of input pins which are 16 pins in this case. The Arduino Mini Pro board that is used in the LMHG contains 14 pins, which is not enough for the purpose of this project. The use of shift registers also means that the LMHG can scale up depending on the need for more input/output pins.

In the initial stage of developing the LMHG, the Arduino code that controls the motors had a code loop that clicks the clock and latch pin 16 times to move the single bit through 16 of the shift register pin positions. At each pin, the enable pin will then be set to high, turning on the motor at each pin position. There is a character array variable that stores 16 values which represents the power of each of the 16 motors. Thus, to change the power of each motor, we change the value of the corresponding item in that array variable. The program listens to incoming serial input data as a single string value coming from the prototype application. The string value is a comma separated value of the 16 power values of the vibration motor array. The string is split and each value is assigned to each item in the power array variable in the Arduino program.

Due to the use of shift registers the vibration motors can only be controlled sequentially, not in parallel. Thus, to change the strength of vibration motor connected via a shift register, the clock pin must click 16 times before coming back to the position of that motor. Normally, the loop which iterates through the shift register happens very fast. Thus, the arrangement of sequential outputs may give the sensation of a parallel or constant output if the output happens too fast for the observer to notice. There is a delay in between the iterations so that the vibration sensation of the motors can be felt. If there was no delay, there would be no observable vibration coming from the motors. Thus the user wearing the LMHG will feel as if the motors are pulsing on and off rapidly.

During the evaluation phase the Arduino code was updated and the delay was removed from the LMHG. In the current version, a *power table* is used to control the power values of each of the motors. This power table works in a way that the variable that stores the motors' power values is iterated through every *power level* in the power table and motors whose power value is equal to or greater than this power level is switched on. Figure 4 below demonstrates how the power table can be used to set the power value of each motor.

Motor power value \ Power level	1	2	3	4	5	6	7	...
1 - power value of 3	ON	ON	ON	OFF	OFF	OFF	OFF	
2 - power value of 6	ON	ON	ON	ON	ON	ON	OFF	
3 - power value of 3	ON	ON	ON	OFF	OFF	OFF	OFF	
4 - power value of 3	ON	ON	ON	OFF	OFF	OFF	OFF	
5 - power value of 4	ON	ON	ON	ON	OFF	OFF	OFF	
...								

Figure 4. How the power table works

In the figure above, the white vertical arrow indicates the process of looping through the power value of each vibration motor and the shaded horizontal arrow indicates the process of looping through the power levels. The power level (horizontal arrow) is incremented by one after a full loop through each motor power value (vertical arrow). The power level loop is repeated when the highest power level is reached. This iterative process happens for the entire duration of the LMHG when it is powered on. For demonstration purposes, only 5 motor power values and 7 power levels are shown. During each loop that goes through the motors' power values, each motor's power value is compared with the current power level. If the motor's power value is less than the current power level, then the motor will turn on and start vibrating. If the motor's power value is higher than the current power level, then the motor will turn off. Therefore, the higher the power value, the longer the motor will continue to vibrate for the duration of the loop that goes through all of the power level. The duration of the vibration creates the impression of a vibration with varying strength. In figure 5, motor 1 has a power value of 3, therefore from the forth power value onwards the motor is turned off. Motor 2 has the highest power value as it only stops vibrating after the sixth power level. This entire process happens very fast and needs no delay in between the iterations like in the previous version of this code. The clock pin is clicked after each motor power value increment and the latch pin is clicked after every power level increment.

The LMHG was initially intended to use the Tactile Brush algorithm (see sub-chapter 3.3) to calculate the vibrotactile feedback. However, due to the hardware setup of the LMHG, the vibration duration and the inter-stimulus onset asynchrony variable (see Israr & Poupyrev 2011) on which the Tactile Brush algorithm depends cannot be created. Also, the Tactile Brush algorithm cannot calculate real-time scenarios (Israr & Poupyrev 2011). Thus, the tactile brush algorithm cannot be implemented in this version of the LMHG.

4.5 Leap Motion integration – the prototype

The final component in the prototype system, after the Leap and the LMHG, is the actual software that will combine these peripherals together and demonstrate their practicability. The chosen theme based on which to develop the software is a file browser. I have chosen to implement a file browser because I want to test use cases that users are most familiar with when using a computer. Browsing files and folders are to me some of the most rudimentary tasks that can be performed on a computer. Figure 5 below is a screenshot of the current version of the file browser application.

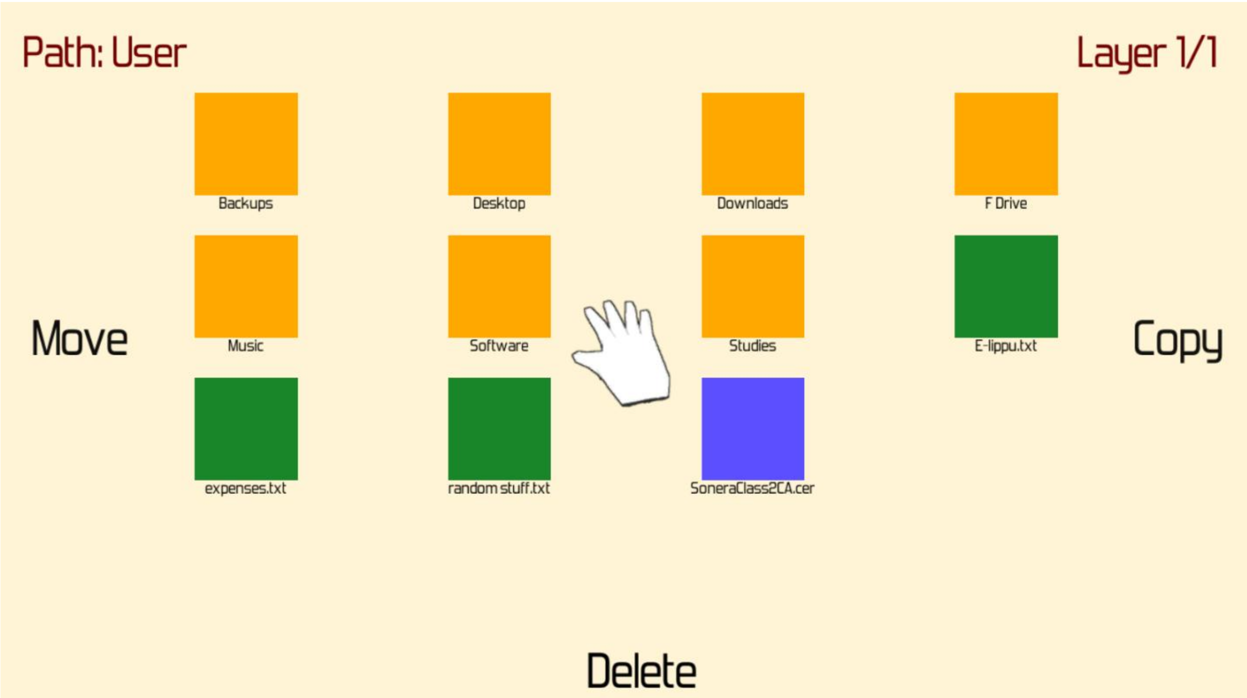

Figure 5. A screenshot of the file browser application

In the figure above, the user's virtual hand is located in the center of the screen. The top text labels tell the user what is the current directory and what is the current layer. The square colored icons represent the folders and files which the user can perform actions such as opening or copying. The “Move” and “Copy” text labels represent the virtual clipboard on which users can place so that they can be moved to a new directory. The “Delete” text label represents a virtual trash can into which users can drag and drop files to delete them.

The challenge is to design a file browser that accepts only gestural input through the use of the Leap. Thus, based on the principles learned in chapter 2, I applied them in order to develop this prototype. The design of the prototype, including the design of the GUI, gestures and interaction styles, and applying haptic feedback is discussed in chapter 5. The figure below demonstrates how the three components of the LMHG system - the prototype software, the LMHG, and the Leap - work together.

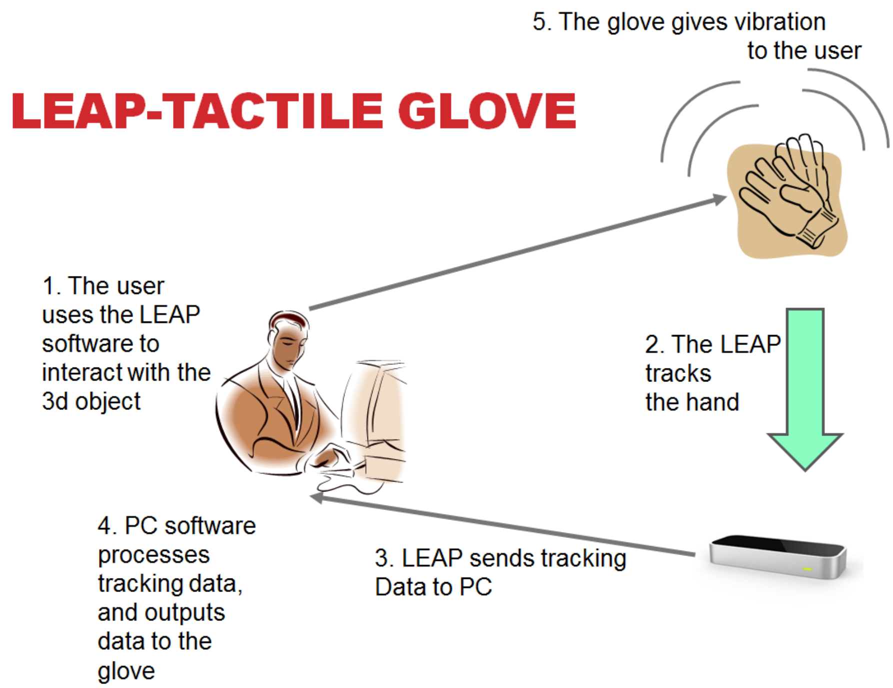

Figure 6. The Leap Motion Haptic Glove prototype system

In figure 6, the order in which events are happening and processed by the three components of the LMHG system is described. As the user is interacting with the software, the Leap tracks the user’s hand, sending the tracking data into the software in order to visualize the user’s hand and simulate virtual interactions with 3D objects in the software. When the user’s virtual hand is interacting with the objects in the software, feedback data is finally sent to the LMHG in order to give the appropriate haptic feedback.

5 Prototype design

In this chapter, I create a framework for designing NUIs that uses touchless interaction enhanced with haptic feedback. This framework is the foundation to build the LMHG system. In addition, new interactions and interface elements are defined to create the prototype, followed by the user interface of the prototype. The steps that I have taken to integrate the prototype software with the LMHG and the Leap are subsequently explained. Lastly, I define the test users and scenarios in which to have users test the LMHG system.

5.1 Framework for building NUIs

Through understanding the live mic problem as discussed in sub-chapter 2.3, I developed a framework to design a touchless NUI that overcomes the live mic problem while adhering to the properties of a NUI, and at the same time incorporates haptic feedback as one of the main feedback modals. Also, using Wigdor and Wixon's framework for the creation of NUIs as a reference (see figure 1 in sub-chapter 2.1), I followed a similar pattern in laying out the steps.

The first step in this framework is to determine the use cases and the context of use. That is, determining the users of the intended NUI, the objectives they are trying to achieve, and the technologies available to create the intended NUI. Understanding the users of the intended NUI means to understand the society in which they live and what cultural and personal traits they are inclined towards. A hand gesture in one culture may have a different meaning in another. An example is the different meanings of the head shake in Western culture and in Indian culture (Norman 2010). After the first step, a series of four different steps that can be iterated is carried out which are to define: the interface elements, the gesture vocabulary, the clutch, and the feedback modals. Evaluation with potential users is necessary as it promotes the iterative development of the interface. Through evaluation, software bugs and poor design choices can quickly be found and fixed. Figure 7 below illustrates the framework to create a NUI with the emphasis on haptic feedback.

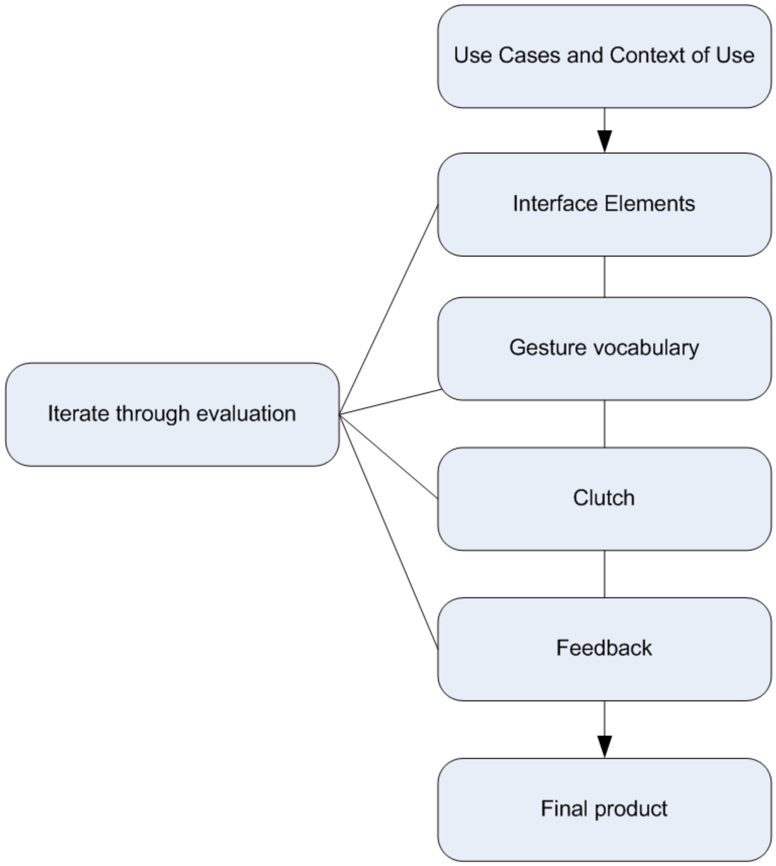

Figure 7. Framework for building touchless NUIs with emphasis on haptic feedback

The interface elements are conceptual entities that have a visual representation in the GUI and can be manipulated by users. In the GUI of current operating systems (OSs) today - such as Windows, Linux distributions, and Mac OS – the interface elements are the buttons, menus, hyperlinks, text boxes, and radio buttons. These elements may seem like the standard components of a GUI, but when designing a touchless interface using mostly gestures, they may not be appropriate due to their intended usage with a mouse or with touch input. Nevertheless, some traditional interface elements can be adapted to be used with touchless interfaces with the appropriate gesture vocabulary to manipulate them.

Defining gesture vocabulary is performed after defining the interface elements. Stern, Wachs, and Edan (2009) defined gesture vocabulary (GV) as the association of a command for an interface to a gestural expression. Thus, defining new GVs involves two steps which are to choose a task dependent set of commands and to choose how to express the commands through gestures. An example is choosing the command volume up for an application, and then choosing a gestural expression that is a clockwise movement of the hand. Stern, Wachs, and Edan (2008) discussed three types of methods that is the specificity method, the rule based method, and the mathematical method. The specificity method means that designers predefine GVs for a system with no explanation for the method used or the choices made. An example research that uses this method is Steptoe and Zhao's (2013) research. The rule based approach is where designers include guidelines to determine the GVs, but do not mention how these guidelines are used to create the actual GVs. An example is the research of Zeng, Sun, and Wang (2012). The last method is the mathematical method, where human factors and other aspects are measured and used to decide which GVs to use. An example of this method is Stern et al. (2009).

The clutch, as mentioned in O'Hara et al. (2014) and Wigdor and Wixon (2011), is a countermeasure for the live mic problem, as discussed in sub-chapter 2.3. Since this framework is specifically intended for touchless interfaces, defining the clutch is a vital step. The clutch can be part of the GVs defined in the previous step, as the clutch can be a gesture that enables/disables other gestures. The clutch should be a gesture that cannot be easily performed as a result of false positive actions and should differ from the other gestures in the GV of an interface. To achieve these characteristics, the clutch gesture should have high recognition accuracy and be comfortable to perform. Stern et al. (2009) lay out formulas to determine the accuracy and comfort level of a gesture. Naturally, all commands that are prone to false-positives should be safe-guarded using a clutch. Also, when defining a gesture for a command, an associated clutch should be defined to accompany that gesture.

The last step to perform is providing feedback to the user. Auditory, visual, and haptic are the modals of feedback that should be consistently used in the design of a NUI. As pointed out by Norman (2010), gestural systems need feedback in order to prompt users of correct and incorrect responses. In this framework, visual, auditory, and haptic feedback should be used in combination and in certain cases emphasizing one modal over another in order to achieve maximum performance from the user. Haptic feedback should be used when auditory feedback is unavailable, or is also not noticeable by the user. Also, haptic feedback should be carefully chosen for certain commands and gestures. If a command is frequently used, having haptic feedback might confuse the user in such a way that they wouldn't understand why that haptic feedback occurred and would choose to ignore it. Feedback to the user should be made for almost all of the gestures in the defined GV.

5.2 Defining new interactions and interface elements

Since I will detach from conventional Windows Icon Menu Pointer (WIMP) designs, defining new interactions and interface elements is the key to exploring the user experience in touchless interaction. Here I take on existing interface elements and manipulate them with new forms of interactions by defining a GV for the prototype.

The existing interface elements which I incorporated in the prototype are icons, the cursor, the clipboard, and text labels. Icons in this context are visual representations of virtual objects of an application. In this file browser prototype, the icons of files and folders are colored squares, and icons of the states of the user's hand are images of a virtual hand. The cursor follows the principle of a mouse cursor which is used to point and indicate selection or focus. The cursor in this prototype is a rectangular object that represents the user's hands. The position of this cursor is determined by the position of the user's hand as sensed by the Leap. The clipboard in the Windows OS is a service where files pending for copy or cut (move) is placed. In this prototype, the clipboard is a visual element in the interface and is shown as a text label with the names of "Copy", "Move", or "Delete" depending on what the users want to do with a file. For example, if the user places a file in the "Copy" clipboard by grabbing and moving to the clipboard area, then that file is shown as visually duplicated and placed visibly on the "Copy", clipboard (this is something the Windows OS does not have). As the user navigates to a directory they wish to paste the copied file to, they simply need to focus on the file in the "Copy" clipboard area and grab, drag, and drop into anywhere in the interface. The text labels are used to display the names of the files and folders and to display various states of the application, such as the current directory, the current page (or *layer* which is the term used in this prototype), name of a clipboard area, and the swipe count. Table 2 below lists the GV of the prototype. An explanation to each of the choices made for each gesture in the GV is given thereafter.

Table 2. Gesture vocabulary of the file browser prototype

Command	Gesture	Feedback – Auditory, Visual, and Haptic
Focus file/folder	Point with 2 fingers	Sound, file/folder enlarges
Open file/folder	Grab and pull	Sound, new layer* loaded
Close file	Swipe 2 times	Sound, current layer reloaded
Deselect file/folder	Swipe 2 times	Sound, file/folder shrinks
Go back a directory	Swipe 4 times	Sound, folder is loaded, whole glove vibrates once
Grab and hold file	Clench hand	File attaches to the hand, whole glove vibrates
Drop file from hand	Open hand	Sound, file flies back to place
Next layer	Grab and pull	Sound, next layer presented
Previous layer	Grab and push	Sound, previous layer presented
Press a button	Hover	Sound, button enlarges
Hover file/folder	Hover	Glove vibrates in finger area**.

*Layer refers to a 2 dimensional plane that the file and folder icons reside.
**The glove only vibrates when hovering over a focused file/folder.

To create a GV for this prototype, first a list of commands should be identified. Since this is a file browser, I have defined the following commands: “open file”, “open folder”, “go back a directory/folder”, “close file”, “choose button”, “next layer”, “previous layer”, “grab file”, “drop file”, “focus on file/folder”, and “stop focusing on file/folder”. To determine the gestures for each of these commands, I followed the rule based approach of determining a GV. In this approach, I lay out a guideline which is to define gestures which follow as closely as possible the corresponding action in the real world for a specific command under a specific context.

For the opening of a file and folder, a grab (clenching the hand) and pull gesture is chosen to mimic as close as possible to the real-life action of picking up an object. In order to prevent false positive, a clutch is needed for this command. This command requires two clutches. I have chosen the gesture of pointing by displaying two fingers, as illustrated in the figure below, to indicate selection or focusing of a file or folder. Only by having focused on an intended file or folder can the user proceed to open them. The finger pointing gesture is the first clutch, because if the user could grab a file without first pointing at it and focusing it, the Leap could at many times falsely read a user gesture and open a random file without the user's intention. The pointing gesture and the state change of the application (a file becomes focused) can together be considered a clutch because it serves as an indicator to start or stop a gesture which in this case is the opening of file or folder gesture. The grabbing action is the second clutch gesture, and the pull gesture that follows is the actual hand motion that the Leap is tracking after the second clutch. Figure 8 below shows the hand gesture for focusing a file or folder.

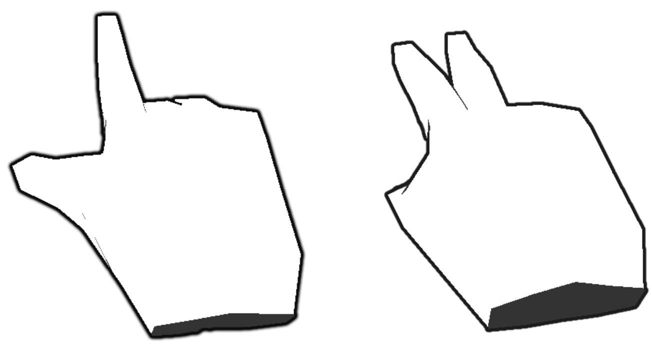

Figure 8. Hand gesture to focus on a file or folder icon

For the "go back a directory" and "close file" commands, I chose a horizontal swipe gesture, where the number of swipes will determine the intended action. Two swipes to "close file" or "stop focusing on file/folder", and four swipes to "go back a directory/folder". The idea behind the horizontal swipe gesture is reminiscent of the act of "shooing" away the unwanted. To perform the "grab file" command, the user needs to actually grab the file icon as they would on a real object. The file icon would then follow the user's virtual hand to indicate that the user is now grabbing that file. Some folders may contain a large amount of files and folders, thus this prototype splits these files and folders into multiple layers that are arranged in depth. This means that if the XY plane is the computer screen view, then these layers are arranged in the Z axis. Figure 9 below illustrates the arrangement of the layers of files and folders in the prototype, and how to use gestures to navigate between them. Notice in the figure that pulling as signified by the red arrow brings the user's vision to the layer behind it (further in the Z axis). The arrow to the left of the figure signifies the movement of the user's vision - in this prototype it is a virtual camera that is viewing the layer - and the color of the arrow indicates in which direction the clenched hand of the user was moving.

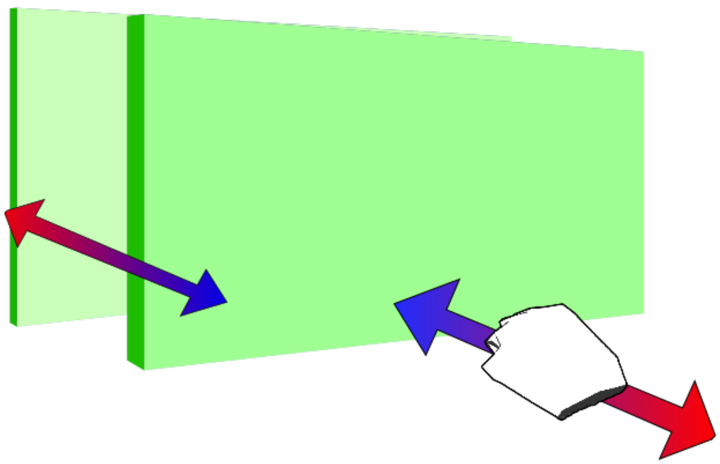

Figure 9. Arrangement of files and folders into layers and changing between layers with gestures

I also imposed the grabbing gesture as mentioned above to enable movement through these layers. Specifically, the user first grabs anywhere on the interface then pulls his clenched hand back towards them to trigger a “next layer” command that moves forward in the above mentioned Z axis. Conversely, a push of a clenched hand indicates a “previous layer” command that moves backwards in the Z axis. The grab gesture in this case is a clutch gesture, because without first having clenched their hand, the push and pull movement of the hand will not be recorded by the Leap and will not trigger any command. The inspiration behind the pull and push grabbing gesture is by suggesting that the user may think of the interface as a slide-able shelf storing files and folders that can be moved by grabbing on to it and either pulling or pushing it in order to access the next or previous shelf.

Lastly, the “Choose button” command applies to the two confirmation buttons that users are prompted with when choosing to delete a file. To press a button, the user hovers their hand over one of the buttons for a specific duration until the button is pressed and a command is subsequently issued.

Visual, auditory, and haptic modals of feedback were used in this prototype. Visual and auditory modals of feedback are apparent in almost every command in this prototype, as shown in table 2. The sounds used in most commands are short beeps which vary in pitch and tone and are easily differentiable. The visual feedback in the commands is composed of transitions and animations to attract the user’s attention. Haptic feedback is apparent only in certain commands supplementing the already shown visual feedback and auditory feedback. Currently, the commands having haptic feedback are hovering over a focused file, grabbing a file, and moving back a directory. When hovering over a focused item, only the finger areas are vibrated and the vibration power is low. When grasping an object, the whole glove vibrates while the file is still being grabbed. When going back a directory, the glove vibrates for a very short period of time.

5.3 Implementing the prototype

The file browser prototype software was developed using the Unity 3D Integrated Development Environment (IDE). The Leap API that was used is the C#-Unity package and was downloaded from the Leap Motion developer website (Leap Motion 2014). To get hand tracking data from the Leap, I referred to the Leap API and various classes, as discussed in sub-chapter 4.2.

Once the hardware for the LMHG was properly set up, I programmed the software to control the LMHG. The software was developed using the Arduino IDE using C. The software listens to incoming input data as a comma-separated values (CSV) string. This string contains the power value of the 16 vibration motors on the LMHG. The input string is split in the software and each power value is assigned to its designated motor power. The power value of each vibration motor takes any value from 0 to 9, with 9 being the strongest vibration power.

After getting the Leap and the LMHG ready for integration, I proceeded to build the file browser software and using the Leap API to create the GV as defined in the last section. The haptic output data (the CSV power values string) is transferred from the file browser to the LMHG through serial connection. The finished file browser software takes data from the Leap to detect the users hand, and then sends haptic data to the LMHG.

6 Evaluation

The purpose of the evaluation phase was to determine the feasibility of using haptic feedback in touchless interaction and to refine the prototype application. The term *participant* is used when referring to the people who took part in this evaluation phase and the term *user* is used in a more general sense, and especially during the analysis and conclusion of this thesis (see Mackenzie 2013, 172). The participant's performance, satisfaction level, and feedback about the overall experience of using touchless interaction with and without haptic feedback served as foundation to determine the feasibility of using haptic feedback in touchless interaction. Each incremental change to the prototype was observed and analyzed through the video recordings done in this evaluation phase in order to provide guidelines to create future NUIs, as well as provide researchers with relevant information to continue research into the topic of this thesis. This chapter outlines the participants, the procedure, and the analysis aspects of the evaluation phase.

6.1 Procedure and tasks

The procedure of the evaluation phase was first to explain to the participants the objective of the thesis, the evaluation conditions, variables, and all the activities the participants could perform during the evaluation and how to perform them. Secondly, a list of the tasks which the participants should carry out was given to them as an ordered list. During this step, I was video recording their actions. Afterwards the participants carry out the tasks. Lastly, the participants were asked a number of questions in the form of a questionnaire to reflect on their experience with the evaluation phase.

A list of tasks was used in the evaluation phase to measure the performance of participants using touchless interaction technologies aided by haptic feedback. The tasks involved activities on a prototype application with the presence of haptic feedback, and subsequently performing similar activities without haptic feedback. When doing the tasks, the participants are recorded on video. The participants were suggested to think aloud (by speaking out loud what they were thinking) to clarify for their actions. By analyzing the video, a couple of performance factors could be measured which is accuracy and stress/comfort (Stern et al. 2009). The evaluation results reached were mostly used to refine the prototype application as well as to improve the framework laid out in sub-chapter 5.1. In addition, the results were also used to address the question of the feasibility of using haptics in touchless interaction.

Two scenarios were carried out in the evaluation: one with the use of haptic feedback, and the other without haptic feedback. The participants were asked to try both scenarios. The activities in the tasks in either scenario were almost identical, except for the ordering of the tasks and certain individual differences (such as file and folder name). This is because participants might be familiar with the procedures after the initial try and will influence the outcome of the remaining scenario's tasks. The tasks were given in a certain context, for example the participants were given a scenario where they are saving their work, or transferring music into flash drives. The evaluation phase was divided so that the first half of participants started the tasks without using the LMHG, and the second half of participants started the tasks using the LMHG.

The questionnaire and the task contents can be found in Appendix A and B.

6.2 Participants

The type of participants for this evaluation phase was determined based on a few criteria. The participant should have basic computer skills, including a basic understanding of files, folders, and the copy, move, and delete operations, which can be performed on the files and folders. Since the prototype uses gestural interaction, knowledge of traditional inputs such as keyboard and mouse was not necessary. Thus, strict age limitations were not applied in this evaluation, although it was preferable that the participants would be experienced enough to use computers on a basic level, and fit enough to do so with reasonable dexterity.

The target number of participants for this evaluation phase was approximately 10-15 people. Due to the circumstances available at this time, I was mostly only able to recruit participants who are my colleagues at the University of Eastern Finland. Also, since the purpose of this evaluation phase was to refine the prototype application, the application version incremented every time a real need for it was observed. Thus, earlier participants worked with the initial versions of the prototype, and later participants worked with the later and more refined versions of the prototype. Since the prototype version was not consistent within the participants, I could not rely on any statistical data analysis. However, since there was a questionnaire that each participant filled out to give their subjective opinion on the prototype, I used the answers of the questionnaire to perform data analysis which is laid out in the next chapter.

The questionnaires given at the end of each test showed that all of the participants were right-handed. There were 10 male and 4 female participants and were all between 18 and 34 years old.

The LMHG is a right-handed glove and as a result this shows that the participants' performances were not affected by the right-handed LMHG. Three participants were experienced with touchless interaction, and the rest either had no experience or little experience.

In terms of ethics, the evaluation phase did not incur any physical or psychological harm to the participants. The participants could quit any time they want to, and the data collected from them were recorded anonymously and no personal detail was disclosed. The participants were given an informed consent form to be signed that states these matters.

6.3 Testing variables, data collection, and data analysis

"An independent variable is a circumstance or characteristic that is manipulated or systematically controlled to a change in a human response while the user is interacting with a computer" (Mackenzie 2013, 161). Independent variables are accompanied by their levels, or test conditions. The independent variable in this evaluation phase is haptic feedback. The levels of this variable are engaging in the interaction with haptic feedback, and without haptic feedback.

A dependent variable is a measured human behavior (Mackenzie 2013, 161). In this thesis, the performance of the participants when being tested, and the overall user experience are the main dependent variables. I measured the participants' performance level by observing the amount of failed and correct gestures through the recorded video of the evaluation sessions.

According to Roto, Law, Vermeeren and Hoonhout (2011), *user experience* can be described as a phenomenon that is a unique experience of a user when using a system and is influenced by their perceptions and prior experiences as well as by the social and cultural context. More specifically, the noun *user experience* denotes an encounter and the experience with a system within a definite time frame. Roto et al. presented 3 factors which affect the user experience which are the context, the user, and the system. The context refers to a mix of social as well as technical context that is separate from the system affecting user experience, the user refers to a user's personal motivation, mood and mental and physical capabilities, and the system is all of the system's properties that influence the user experience. In the evaluation of this research's prototype, the factor that is taken into consideration when examining the participants' user experience is mainly the system factors. Specifically, the user experience that was analyzed in this research is concerned with the interface of the prototype application and the haptic feedback that is provided through the LMHG when using the application. The user factor such as mood and personal motivation and the context factor such as using the application in a group are not taken into account.

There are two main types of data collected in this evaluation phase. First, real-time data was taken, taken through video recording of participants' behaviors, mistakes, and anomalies. These videos were later analyzed in detailed. Answers from the questionnaires were the second type of data collected.

The video data was compiled by measuring different types of gestures which occurred during the evaluation sessions and also taking notes of any points of interest such as abnormal user behavior or the high frequency of certain user action. The questionnaire data was compiled by coding the text and categorizing key terms and concepts. After compilation, this data was analyzed to draw recommendations and good practices for the prototype application, and in turn give practical guidelines for building touchless gestural interfaces. The data was also compiled so that an understanding could be reached on the use of haptic feedback in touchless interaction. Specifically, the compilation of video recording data was used to measure the performance level of participants in both cases of wearing the LMHG and without wearing it. This performance level could subsequently give indications on the impact of having haptic feedback in touchless interaction. This section describes the compilation process of the video recording and the questionnaire, the subsequent data analysis on the compiled data, and the resulting consensus and understanding of the thesis topic through analysis. While laying out the results, I will also discuss how the versions of the prototype were incremented and how the evaluation phase helped develop a more responsive and well-designed prototype software.

In the next chapter, I will apply knowledge gained from the background research, as well as the data gathered from the evaluation phase in order to analyze the data.

7 Results

A total of 14 participants were tested on the prototype file browser application. There were 4 iterations to the development of the prototype, resulting in 5 participant groups who tested the different versions of the prototype. 12 of the 14 participants were tested in two testing scenarios, one with the use of the LMHG, and the other one without. The remaining 2 participants tested in only the scenario without the LMHG. Hereafter the term *scenario* means a testing session which only tests one of the two scenarios mentioned above. Therefore, the total number of testing scenarios is 26. Insight into how the incremental changes in the prototype affected the participants' performance, experience, and feedback is discussed in this chapter.

7.1 Test data analyses

To measure the performance level of participants in the test, I counted the total number of gestures of the scenarios – with and without the LMHG – of each participant. My basis for this measurement of performance was due to the assumption that the least amount of gestures means that the participant showed good performance when performing tasks. On the contrary, a participant having high gesture counts may indicate that they are having a difficult time performing the tasks, and therefore their performance is lowered. I also counted different types of gestures: correct gestures, wrong gestures, and false positive gestures which together made the total gesture count. Wrong gestures included both gestures that are not in the GV, and gestures which looked correct but does not issue any command in the application. False positives were gestures which the user unintentionally made that the application registered as valid and proceeded to issue the appropriate command. Thus the user was usually confused as to what happened when a false positive gesture occurred. Note that the total gestures include all gestures that I have observed, even if the gestures were not part of the task. The remaining type of gesture, other than wrong and false positive gestures, is correct gestures which are gestures that are successfully performed to issue a command that the user intended.

The prototype system went through six major milestones. The first milestone was a result of a high wrong gesture count when participants were trying to grab a file. The Leap provided an inaccurate reading of the open/closed state of the hand. Therefore to make it easier for the participants, a new feature was added to the file browser software that the user could grab a file in focus regardless of the position of the hand. Although this feature helped lower the wrong gesture count, it was later removed in milestone 5 because participants were not aware of this feature and also some participants suggested that it was more natural for them to grab a file only when their virtual hand was hovering over it. The second milestone removed a feature that cleared the file clipboard when a two-swipe gesture is detected. Due to the high false positive rate of the swiping gesture, the participants had to redo some tasks because this feature undid their progress. Although after this milestone, the remaining participants' total gesture count did not lower as expected. This is because the participants still suffered from other false positives and wrong gestures.

The third milestone created an indicator to let participants know how many swipes they have made. This feature was important because the majority of false positives and wrong gestures were comprised of swipe gestures. Despite having these features, only a couple of participants noticed this indicator. The fourth milestone arose after feedback from a participant suggesting that the LMHG vibrated constantly, thus giving no clue as to why the vibrations occurred. This milestone changed the haptic feedback so that the LMHG only gives off vibrations when the virtual hand is hovered over a focused file or folder. This is because the majority of the time the participants were hovering over either a file or folder.

The fifth milestone fixed a couple of major bugs that accounted a high number of wrong and false positive gestures. The only observable increase in performance is seen after the fifth milestone where the software bugs have been fixed and in the sixth milestone where the delay of the LMHG was reduced significantly. The performance of the last three participants improved as expected, although not significantly because they had difficulty in comprehending and performing the tasks. Five of the six evaluation sessions of these three participants were lower than the mean value, indicating the improvement in performance. The evaluation phase showed that even with a limited number of test users, rapid incremental development of the prototype can be done and that major bugs can be detected and fixed and essential features to the prototype can be added.

Figure 10 below shows the gesture counts of all the participants' gesture types and the milestones which the prototype went through.

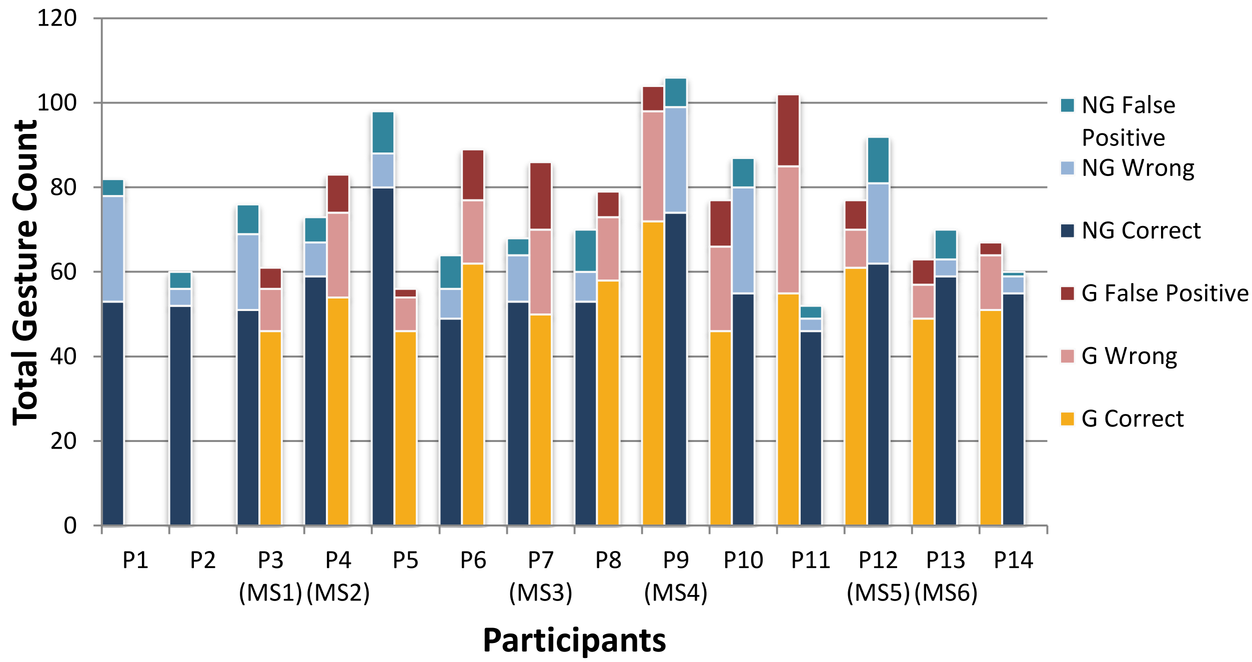

Figure 10. A clustered, stacked column graph displaying the gesture count of participants

In figure 10, the abbreviations are as follows: P stands for participant, MS stands for milestone, G stands for glove which means the scenarios with the LMHG, and NG stands for no glove which means the scenarios without the LMHG. The timeline for the milestones is determined based on the order of the participants that were tested. For example, "P4 (MS 2)" indicates that milestone 2 is taken into effect from the fourth participant. Due to technical difficulties experienced with the first two participants, the video recordings were cut short and I was not able to count the gestures for the scenarios that used the LMHG. Except the first two participants, each participant underwent 2 testing scenarios represented by 2 columns. The column in shades of blue represent the gesture counts of the scenario with no LMHG, and the column in shades of red and yellow represent the gesture counts of the scenario with the LMHG. Two mean values of the total count of gestures were calculated for the 2 scenarios. The mean value for the scenario with the use of the LMHG is 79 gestures, and the mean value when not using the LMHG is 76 gestures.

Of all the tests that used the LMHG, 6 out of 12 or 50% of the tests were below the mean value for this scenario. 8 out of 12 or 67% of the scenarios that did not use the LMHG were below the corresponding mean value. The mean total gesture count of each scenario gives an indication of the overall performance level of all the participants. Participants below the mean threshold generally performed better than the other participants, and in general performed the given tasks well. The percentages specified above of the participants' performances that are below the mean values in the 2 scenarios indicate that in general the participants performed slightly better with bare hands than with the use of the LMHG.

I calculated 2 threshold values to which to compare the results which are perfect execution and best execution. A perfect execution is a test session where the tasks are done with the least amount of gestures. The value for a perfect execution is the total gesture count of a preliminary evaluation session that I conducted by myself where no mistakes are made and no redundant steps are taken to complete the tasks. I took this preliminary evaluation session to serve as a baseline for the participant evaluation sessions thereafter. The best execution is the lowest total gesture count of the 12 participants. The best execution is determined by adding the lowest gesture count of each gesture type. The further away from the best execution values, the lower the performance of a participant is. Table 3 below lists the calculated standard deviation of the scenarios and gesture types.

Table 3. Standard deviations and mean values of the scenarios and gesture types

Gesture types / Values	Total gesture count	Correct	Wrong	False Positives
Mean **with** haptic glove	78.7	54.2	16.2	8.3
Mean **without** haptic glove	76.3	58.0	11.6	6.8
Standard deviation **with** haptic glove	14.7	7.6	6.9	4.6
Standard deviation **without** haptic glove	16.1	9.6	7.7	2.8

The standard deviation indicates to which degree the values differ from each other. For example, the standard deviation of the total gesture count with the LMHG is 14.7, and when compared to the mean of 78.7 it means that the participants who used the LMHG had quite varying performance levels. The tests that did not use the LMHG had very similar values and also indicated that the performance levels in this scenario were not so close to each other. The standard deviations of the wrong gestures and the false positive gestures in both scenarios are approximately 40-60% of their corresponding mean values. This occurrence is related to the fact that some participants experienced high frequencies of software bugs and poor hand tracking performances while some experienced very low frequencies.

Out of the 15 tests that are below the mean in terms of total gesture count, sessions that do not use the LMHG comprises 9/15 or 60%. This means that the performance level of participants when not using the LMHG is generally higher than when the participants used the LMHG. A closer examination of the gesture count of each participant in the two scenarios show that sometimes the participants performed better (lower gesture count) when not using the LMHG, and vice versa. This occurrence is also affected by the order in which the scenarios occurred because the gesture count in the second scenario could be lower than the first scenario due to learning. Due to this learning effect, I planned the evaluation so that the first half of the participants was to perform the scenario without the LMHG first. The remaining half performed the scenario with the LMHG first. The raw gesture count data collected from the video recordings alone indicate that it is better to use bare hands to use touchless interaction than a haptic glove with the given evaluation conditions. Thus, the performance level of the participants did not improve when using the LMHG.

After observing the video recording, I have identified a number of factors that contribute to the user's performance: not reading the task clearly resulting in doing the wrong actions, costly false positive gestures which hinder a participant's usage in such a way that it makes the participant repeat their actions, and the tracking device not correctly tracking the hand status (mainly finger count and swipes) which caused wrong gestures resulting in more gesture count to complete the task list.

The main causes of user errors, and as a result a reduction in their performances, are due to the performance of the tracking device and the maturity of the application being used. The Leap showed erratic tracking behavior and the software bugs were encountered during evaluation. As a result, there was a high false positive count on swiping gestures and grabbing and releasing gestures. Most of the wrong gesture count values seen in figure 10 are also due to the failure to recognize these gestures. A few participants had difficulty following the task which also attributed to the reduced performance. The tracking device is hard to fine tune, especially in this research as the Leap is not the main focus. For this reason, evaluating and refining the prototype application is essential to increase a user's performance.

The video recording data and the prototype milestones indicated that although through each milestone several features were added and removed to help the participants achieve higher performance, the increment of the milestones helped little with the user's performance. At this point a conclusion can be reached is that that the LMHG does not increase the user's performance when using touchless interaction. This is because the factors that contribute to a user's performance did not improve with the use of haptic feedback. Moreover, a look back into figure 10 shows that even after milestone 5 where most of the bugs have been fixed, 2 of the remaining 3 participants still performed better without the LMHG. This shows that the recognition of the Leap when using the LMHG may be another factor affecting performance. Nevertheless, the video recording data alone may have not presented enough data to determine whether or not haptic feedback improved the participants' performance.

The questionnaires (see appendix A) that followed each video recording give indications of the participants' characteristics and their opinions on the prototype. Despite having or not having prior experiences with touchless interaction, all of the participants thought that the file browser application was easy to learn. The questionnaire asked the participants how satisfied they were in the two scenarios of the test, and figure 11 below shows the changes in satisfaction level in the timeframe of the evaluation using a line graph. The level of satisfaction ranges from being very unsatisfied (0) to very satisfied (4).

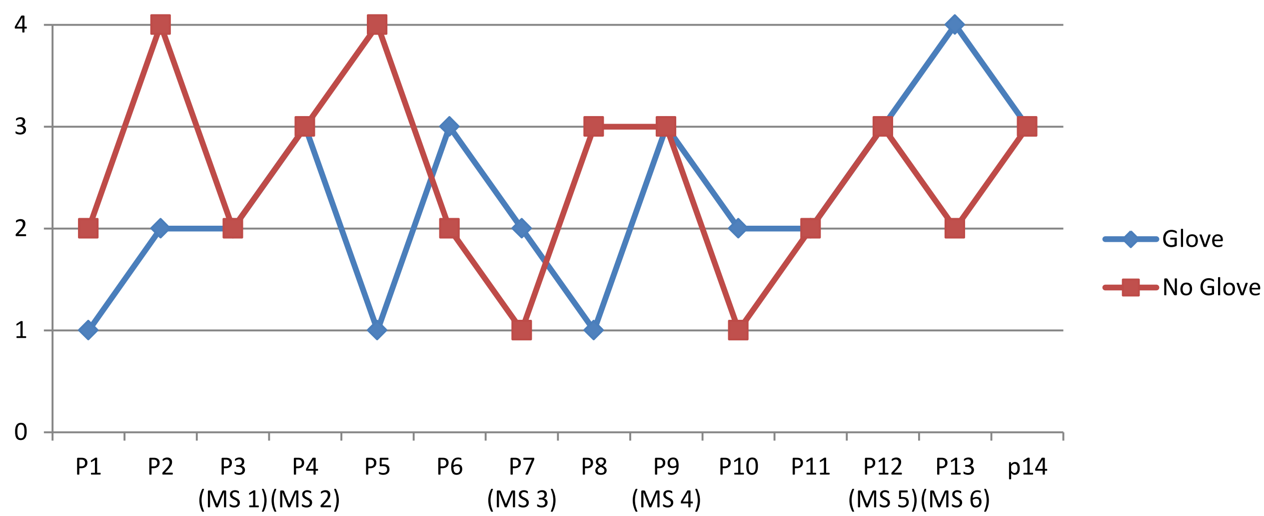

Figure 11. The participants' reported satisfaction level throughout the evaluation

The satisfaction level of the participants shows how much they enjoyed the experience of using the prototype, as well as how comfortable they felt when performing the tasks. This explanation of the satisfaction level was not explained in the questionnaire, but was given verbally to the participants. Although the video recording data of the first two participants in the scenario with the LMHG was absent, the participants still gave a satisfaction level to the experience. In figure 11, the satisfaction level of participants in the scenario that used the LMHG has an overall upward trend throughout the evaluation period. There was a fluctuation in satisfaction levels between milestone 2 and milestone 4 when participants used the LMHG which may be attributed to the random occurrence of software bugs and the delay in the haptic feedback. The satisfaction level when not using the LMHG was generally high with only 2 participants reporting low satisfaction between milestone 3 and 4. After milestone 5, the major bugs were fixed and in milestone 6 the delay in the haptic feedback was gone which saw the rise in satisfaction level in both scenarios. The bug fixes, reduced delay of the LMHG and other enhances to the prototype made the interaction experience easier and have less frustration. To sum up figure 11, the participants' satisfaction levels rose after the software bugs were fixed and the LMHG was enhanced. There is not enough data at this stage to determine if the performance of the participant affected their satisfaction level or not.

The questionnaire contained 4 free-form questions that asked about the user's preference of using the LMHG, their feedback on the use of the interface elements and GV of the application, and their opinions on the use of haptic feedback in touchless interaction in general.

The participants had mixed opinions about using the LMHG. The first question asked whether or not the participants preferred using the LMHG and the results are that 6 people would rather use bare hands, 4 people were more comfortable with the LMHG, and the remaining 4 people were undecided. The reasons for preferring to use bare hands were varied, for example the glove was uncomfortable to wear, the delay of the haptic feedback, and the erratic behavior of the haptic feedback. Of the participants who preferred the LMHG, 3 said that the LMHG enhanced the experience of the interaction, while the remaining participant emphasized that the LMHG helped them understand better the state of the application. Of the participants who were undecided, some justified their answer due to the fact that they were unaware of the LMHG in their hand for the duration of the test. The remaining participants either had no reason or suggested that the technology of the LMHG should be made portable and more elegantly designed.

Two free-form questions asked the participants about what aspects of the interaction and the interface elements of the prototype application they liked the most and what they think needed improvements. When asked about what the participants liked the most, the most popular responses were related to the GV which include the act of opening files and folders, followed by the grabbing motion, the swiping gesture, and the act of focusing files and folders. Other elements which were mentioned included the copy, move, and delete operations, the action of moving back and forth between layers, and hovering over buttons. An opinion was given that the tracking device was very accurate. This opinion was surprising due to the fact that the poor tracking of the Leap was one of the factors causing low user performance.

The aspects of the prototype which needs improvement according to the participants include the poor accuracy and limitation of the tracking device, followed by the action of focusing items and the swiping gestures. These top mentioned aspects that need improvement were the reasons for the high number of wrong gestures and false positive gestures. Despite the participants listing the same gestures that they gave for aspects which they liked in the previous question, the participants mentioned that it was not the gesture itself that they thought needs improvement but rather due to the way the Leap tracked these gestures. What can be inferred from these responses is that the gestures which needed improvement were hard to track by the Leap and should be redesigned either by updating the software code or by considering an alternative way to track the gestures. Other than gestures and the poor tracking of the Leap, the participants also listed the delay and erratic behavior of the haptic feedback and the choice of using haptic feedback in certain commands. Moreover, a couple of suggestions were given that there should be clearer feedback of certain actions because some actions happened too quickly or was not so clear to the participants. Regarding the prototype in general, one response was that using the prototype made the participant's arm sore because their arm was not resting on any surface. Other suggestions were that the interface should have fewer elements because it looked cluttered and that there should be a feature to select multiple files at the same time.

The last free-form question is an open-ended question that asks the participants' opinions on haptic feedback in touchless interaction in general. Overall, the responses suggested that since haptic feedback makes use of the human sense of touch it can be used both in recreational purposes as well as being an effective assistance technology for the elderly and handicapped. One response suggested that the haptic feedback could be used in an online shopping application where a user could feel the products' texture with the use of a haptic glove.

7.2 Feasibility of haptics in touchless interaction

Based on the videos and questionnaire responses of the 14 participants, I have identified four issues that affect the feasibility of using haptic feedback in touchless interaction: (1) the attention and frustration level of the user, (2) the quality of the devices enabling tracking and feedback, (3) the understandability and learnability of the GV, and (4) the personal experience of the user regarding the haptic feedback elements.

The attention level shows how aware the user is of the state of an application and the frustration level measures how much they are struggling when using that application. To design a good interface, the attention level of the user should be raised, and the frustration level should be reduced. In order to do so, the interface elements should provide efficient and effective feedback. Providing efficient feedback means that the use of feedback modalities is balanced and logical and providing effective feedback means that the feedback modalities used should have a clear impact on the user. For example, I observed that when a false positive action occurs and the action happened too quickly for a visual feedback to become noticeable to the user, the user was still prompted with auditory and haptic feedback that told the user whether the action happened. Although in this example the user was distracted by the false positive, the frustration level of the user remained low because they were still aware of what had happened due to the efficient feedback. The false positive gestures in this example were the cause of the lowered attention level. The false positive gestures can be mitigated by evaluating the interface and fixing software bugs or by studying other contributing factors such as tracking device, surrounding environment, or similarity among gestures to determine the cause of the false positive gestures. During initial testing with participants, I came to know that too much haptic feedback confused the user in that they were unable to tell which commands generate what kind of vibration. As a result, this reduced the attention level of the user and raised their frustration level.

The quality of the devices used for the tracking of the user's bodily gestures directly impacts the user's performance. As can be seen from the data analysis in the last section, the causes of poor user performance were mostly due to the erratic tracking behavior of the Leap. The erratic tracking behavior was the cause of high false positive and wrong gesture count in case of many participants. Since touchless interaction depends on the tracking device, the quality of the device limits the quality of the interfaces which are built for these devices. Unless a study is focused on creating or improving a tracking device, the innate quality of a tracking device cannot be improved. Consequently, studies that use tracking devices to carry out research should evaluate their quality and performance as the results of the research may depend on it. Devices which give feedback to the user also have an impact on the user's performance and attention and frustration level. Deriving from the studies of haptic feedback discussed in sub-chapter 3.2 and the results of the evaluation phase analyzed in the last section, the user experience of an interface that uses haptic feedback is affected by the haptic feedback device. Therefore, to enhance a NUI with haptic feedback, a responsive and accurate device providing haptic feedback is essential.

The understandability and learnability of the GV affect the user's performance level when using an interface. In the evaluation, when the participants did not remember the associated gesture for a command, they tried to guess it and as a result created wrong gestures as well as false positive ones. Moreover, in some of the participants evaluated, the random occurrence of haptic feedback confused their interaction experience and consequently they had a difficult time distinguishing gestures and commands. If the haptic feedback had behaved exactly as described in the GV defined in sub-chapter 5.2, the participants' performance could have increased further. Based on the feedback given by the participants who evaluated the prototype after the sixth milestone, the performance increase could be due to the participant's association of the haptic feedback to the associated commands in the GV.

The user's personal experience of haptic feedback elements indicates how the user interprets the haptic feedback. When the prototype was still in the early milestones, the erratic behavior of the haptic feedback confused the participants and as a result each participant had their own interpretation of what the haptic feedback meant. For this reason, the feedback should be clear to the user and should not be interpreted differently between different users. In order to create meaningful haptic feedback, the haptic feedback types that are used should be considerably different from each other and should be tested with users to refine the haptic feedback. In the GV of the prototype of this research, the haptic feedback types that were tested were soft vibration on the fingers, strong vibration on the whole hand, and short-lived vibration on the whole hand.

Deriving from the discussions of the four issues above, the quality of the devices used for the tracking gestures and the devices that give haptic feedback affect the user's attention and frustration level, their ability to remember and effectively use the GV, and their personal experience of haptic feedback elements. Moreover, if the user is distracted by anomalies in the interface and their frustration level is high, then their ability to remember and effectively use the GV would decrease. Failing to utilize the GV, the user's experience of any associated haptic feedback would in turn become unsatisfactory.

A definite conclusion to this research is not reached, as the data collected is but a small sample of the many ways touchless interaction is used. Haptic feedback may be applicable in this scenario, but not in others. Nevertheless, the four issues identified and analyzed above determine that it is feasible to provide haptic feedback to touchless interaction but only when these issues are understood and the principles that they convey are applied when creating touchless interfaces using haptic feedback.

8 Conclusion

Touchless interaction provides means to create NUIs which are changing the way humans interact with computers. Most touchless interaction applications discussed in this thesis work lack the haptic feedback that is apparent in touch-based interaction. Even by definition, touchless interaction should not involve any form of contact with the physical system. In a practical sense, touchless interaction involves the user to perform gestures using their bodies which touchless interaction devices track. These gestures in turn provide input for manipulating the touchless interface. The haptic technology that is studied in this research refers to the computer subfield that deals with the development of systems that create a virtual sense of touch. The haptic technology that this research studied involves simulating cutaneous input, which means sensations of the skin, using vibrotactile feedback created with vibration motors embedded in a haptic glove.

Touchless interaction has the benefit of providing a sterile environment and promotes freedom of movement, amongst other benefits that derives from the touchless nature of this interaction. The lack of haptic feedback and the live mic problem are shortcomings of touchless interaction. Motivated by the need to address these shortcomings, this thesis work studied how haptic feedback can be integrated into touchless interaction. With this intention, this thesis work studied the background of touchless interaction and haptic technology, and developed a framework for building touchless interfaces using haptic feedback. This framework was further strengthened by the creation of a prototype application that uses touchless interaction with a haptic feedback glove and an evaluation phase that followed.

The prototype consisted of a file browser application that uses hand gestures to manipulate files and folders. This application used the Leap Motion controller which tracks the user’s hands and fingers, and the haptic feedback glove – the LMHG – which provided haptic feedback by means of vibrotactile feedback. The evaluation phase that followed the initial version of the prototype was conducted to improve on the prototype, as well as to determine whether or not haptic feedback in touchless interaction was feasible.

The data in video recordings and questionnaires from the evaluation phase showed that there are four issues to consider for haptic feedback to be feasible in touchless interaction. For haptic feedback to work with a touchless interface, the quality of the device providing touchless interaction and the device providing haptic feedback should be high, and the GV of the interface and the associated haptic feedback should be well designed to keep the user's attention level steady and their frustration level minimal. If the quality of the device providing touchless interaction is low, any positive effects given by the haptic feedback device may be overlooked. Moreover, when designing a touchless interface using haptic feedback, the haptic feedback should be clear and well-defined to keep the user focused on using the application and should be used sparingly so that the user is not overwhelmed by it.

Haptic feedback integration into touchless interaction may see benefits in certain contexts of use. In clinical environments where sterility is vital, and in environments where the system must be kept out of contact from users, such as public systems, the haptic glove is not feasible. However, in other situations, such as learning and assistive environments, entertainment settings, robotics control systems, and home environments, the haptic glove may indeed prove to become beneficial.

This research has shown that haptic feedback is possible in touchless interaction in a specific context of use. With the frameworks given in this research, as well as the analyses of data gathered from the evaluation phase, the continuation of this research topic may be useful in confirming the potential benefits of haptic feedback in touchless interaction. Moreover, further research into the topic of this research may explore how haptic feedback can be used to build effective assistance applications for the elderly and handicapped, as well as applications for learning and recreational purposes.

References

Abrahamsson, Pekka & Salo, Outi & Ronkainen, Jussi & Warsta, Juhani 2002. Agile software development methods – Review and analysis. VTT Publications, Espoo.

Airspace Store 2014. Leap Motion, Inc.
<https://airspace.leapmotion.com>

Bartoli, Laura & Cooradi, Clara & Garzotto, Franca & Valoriani, Matteo 2013. Exploring motion based touchless games for autistic children's learning. Proceedings of the 12th International Conference on Interaction Design and Children. 102-111.

Becker, D.A. and Pentland, T. 1996. Staying alive: A virtual reality visualization tool for cancer patients. In Proceedings of the AAAI Workshop on Entertainment and Alife/AI. AAAI Technical Report WS-96-03.

Camara, Antonio 2011. Natural User Interfaces. Keynote speaker in Campos, Pedro & Graham, Nicholas & Jorge, Joaquim & Nunes, Nuno & Palanque, Philippe & Winckler, Marco (Eds.) Human-Computer Interaction – INTERACT 2011, Springer. 1.

Cassidy, Brendan & Cockton, Gilbert & Coventry, Lynne 2010. A Haptic ATM Interface to Assist Visually Impaired Users. Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility.

De-La-Barre, Rene & Chojecki, Paul & Leiner, Ulrich & Muhlbach, Lothar & Ruschin, Detlef 2009. Touchless Interaction: Novel Chances and Challenges. Proceedings of the 13th International Conference on Human-Computer Interaction. 161-169.

Educypedia 2013. Shift registers. Viewed on 29th April 2014.
<http://educypedia.karadimov.info/library/dns_module3_p3.pdf>

Etherington, Darrell 2013. Leap Motion Controller Ship Date Delayed Until July 22, Due To A Need For A Larger, Longer Beta Test. Read on 29th April 2014.
<http://techcrunch.com/2013/04/25/leap-motion-controller-ship-date-delayed-until-july-22-due-to-a-need-for-a-larger-longer-beta-test>

Fikkert, Wim & Vet, Paul van der & Nijholt, Anton 2010. User-evaluated Gestures for Touchless Interactions from a Distance.

Gallotti, Paulo & Raposo, Alberto & Soares, Luciano 2011. V-Glove: A 3D virtual touch interface. Proceedings of the XIII Symposium on Virtual Reality, IEEE. 242-251.

Ghiani, Guiseppe & Leporini, Barbara & Paterno, Fabio 2009. Vibrotactile feedback to aid blind users of mobile guides. Journal of Visual Languages & Computing, Vol. 20, Issue 5. 305-317.

Israr, Ali & Poupyrev, Ivan 2011. Tactile Brush Algorithm. CHI, ACM.

Joppe, Marion 2014. Exploratory Research. The Research Process. University of Guelph, Canada.

Kaczmarek, Kurt A. & Webster, John G. & Bach-y-Rita, Paul & Tompkins, Willis 1991. Electrotactile and Vibrotactile Displays for Sensory Substitution Systems. IEEE Transactions on biomedical engineering, Vol. 38, No. 1. 1-16.

Karam, Maria & Schraefel, m. c. 2005. A Taxonomy of Gestures in Human Computer Interactions. University of South Hampton.

Kim, Sang-Youn & Kim, Kyu Yong & Soh, Byung Seok & Yang, Gyunghye & Kim, Sang Ryong 2006. Vibrotactile Rendering for Simulating Virtual Environment in a Mobile Game. IEEE Transactions on Consumer Electronics, Vol. 52, No. 4. 1340-1347.

Kosner, Anthony Wing 2013. Leap Motion Announces $30M Series B Funding And Bundle Deal With ASUS Computers.

Kuno, Yoshinori & Murashima, Teruhisa & Shimada, Nobutaka & Shirai, Yoshiaki 2000. Intelligent Wheelchair Remotely Controlled by Interactive Gestures. Proceedings of the 15th International conference on Pattern Recognition. 672-675.

Leap Motion 2014. Leap Motion Developer Website. <https://developer.leapmotion.com>

Lederman, S. J. & Klatzky, R. L. 2009. Haptics perception: A tutorial. Journal on Attention, Perception, & Psychophysics, Vol. 71, Issue 7. 1439-1459.

Mackenzie, Scott I. 2013. Human-Computer Interaction: An Empirical Research Perspective. Morgan Kaufman, Elsevier.

Malizia, Alessio & Bellucci, Andrea 2012. Viewpoint – The Artificiality of Natural User Interface. Communications of the ACM, Vol. 55, No. 3. 36-38.

MarketWired 2012. Leap Motion Announces $12.75 Million Series A Funding Round Led by Highland Capital Partners. Read on 29th April 2014. <http://www.marketwired.com/press-release/leap-motion-announces-1275-million-series-a-funding-round-led-highland-capital-partners-1654737.htm>

Morrell, John & Wasilewski, Kamil 2010. Design and evaluation of a vibrotactile seat to improve spatial awareness while driving. Department of Mechanical Engineering, Yale University. Proceedings of IEEE Haptics Symposium. 281-288.

Nishikawa, A & Hosoi, T. & Koara, K. & Negoro, D. & Hikita, A. & Asano, S. & Kakutani, H. & Miyazaki, F. & Sekimoto, M. & Yasui, M. & Miyake, Y. & Takiguchi, S. & Monden, M. 2003. FAce MOUSe: A novel human-machine interface for controlling the position of a laparoscope. IEEE Transactions on Robotics and Automation, Vol. 19, Issue 5. 825 -841.

Norman, Don 2010. Natural User Interfaces Are Not Natural. ACM Interaction Design, Vol. 17, Issue 3. Read on 8th March 2014. <http://www.jnd.org/dn.mss/natural_user_interfa.html>

Norman, Don 2013a. Opportunities and Challenges for Touch and Gesture-Based Systems. The Next Touch Evolution. Advancing the Consumer Experience in Other Realms: Tasks and Tough Environments. Society for Information Display conference, Vancouver, Canada. Read on 9th March 2014. <http://www.jnd.org/dn.mss/opportunities_and_ch.html>

Norman, Don 2013b. The Design of Everyday Things, Revised and Expanded. New York; London: Basic Books; MIT Press (British Isles only).

Nguyen, Viet T. 2013. IT Project Report. Interactive Technologies, University of Eastern Finland, Joensuu.

O’Hara, Kenton & Harper, Richard & Mentis, Helena & Sellen, Abigail & Taylor, Alex 2012. On the naturalness of touchless - putting the “interaction” back into NUI. Microsoft Research, Cambridge, UK. ACM Transactions On Computer Human Interaction.

O’Hara, Kenton & Gonzalez, Gerardo & Sellen, Abigail & Penney, Graeme & Varnavas, Andreas & Mentis, Helena & Criminisi, Antonio & Corish, Robert & Rouncefield, Mark & Dastur, Neville & Carrell, Tom 2014. Touchless Interaction in Surgery. Communications of the ACM, Vol. 57, No. 1. 70-77.

Placitelli, Alessio Pierluigi & Gallo, Luigi 2012. Toward a framework for rapid prototyping of touchless user interfaces. Sixth International Conference on Complex, Intelligent, and Software Intensive Systems. IEEE. 539-543.

Regenbrecht, Holger & Hauber, Joerg & Schoenfelder, Ralph & Maegerlein, Andreas 2005. Virtual Reality Aided Assembly with Directional Vibro-Tactile Feedback. 1-59593-201-1/05/0010 ACM. 381-387.

Robles-De-La-Torre, Gabriel 2006. The Importance of the Sense of Touch in Virtual and Real Environments. Haptic User Interfaces for Multimedia Systems IEEE. 24-30.

Robles-De-La-Torre, Gabriel 2010. International Society for Haptics: Haptic technology, an animated explanation.

Rodriguez, Salvador 2013. Leap Motion's 'Free Form' app is like Play-Doh of the digital age. Read on 29th April 2014. <http://www.latimes.com/business/technology/la-fi-tn-leap-motion-3d-free-form-playdoh-20131120,0,3251169.story#ixzz30DiJiXCp>

Roto, Virpi & Law, Effie & Vermeeren, Arnold & Hoonhour, Jettie (Eds.) 2011. USER EXPERIENCE WHITE PAPER: Bringing clarity to the concept of user experience. Result from Dagstuhl Seminar on Demarcating User Experience 2010. Read on 23rd May 2014. <http://www.allaboutux.org/files/UX-WhitePaper.pdf>

Shrewsbury, Brandon 2011. Providing Haptic Feedback Using the Kinect. ASSETS’11, ACM. 321-322.

Spano, Lucio Davide 2012. Developing Touchless Interfaces with GestIT. Lecture Notes in Computer Science Vol. 7683. 433-438.

Steptoe, William & Zhao, Lu 2013. Poster: Using the Whole Body for Multi-Channel Gestural Interface. Dept. of Computer Science, University College London. IEEE Symposium on 3D User Interfaces.

Stern, Helman & Wachs, Juan & Edan, Yael 2008. Optimal Consensus Intuitive Hand Gesture Vocabulary Design. Proceedings of IEEE International Conference on Semantic Computing. 96-103.

Stern, Helman & Wachs, Juan & Edan, Yael 2009. A Method for Selection of Optimal Hand Gesture Vocabularies. Lecture Notes in Computer Science, Vol. 5085. 57-68.

Szczys, Mike 2011. How shift registers work. Viewed on 29th April 2014. <http://jumptuck.com/2011/11/02/how-shift-registers-work/>

Tsotsis, Alexia 2011. OcuSpec Raises 1.3M From Andreessen And Others To Build An "Affordable Kinect". Read on 29th April 2014.

<http://techcrunch.com/2011/06/10/ocuspec-raises-1-3m-from-andreessen-and-others-to-build-an-affordable-kinect>

Van Der Linden, Janet & Johnson, Rose & Bird, John & Rogers, Yvonne & Schoondervaldt, Erwin 2011. Buzzing to Play: Lessons Learned From an In the Wild Study of Real-time Vibrotactile Feedback. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 533-542.

Wachs, Juan Pablo & Stern, Helman & Edan, Yael & Gillam, Michale & Feied, Craig & Smith, Mark & Handler, Jon 2008. A Gesture-based Tool for Sterile Browsing of Radiology Images. Journal of the American Medical Informatics Association. Vol. 15, Issue 3. 321 -323.

Wachs, Juan Pablo & Kölsch, Mathias, & Stern, Helman & Edan, Yael 2011. Vision-Based Hand-Gesture Applications. Communications of the ACM, Vol 52, No. 2. 60-71.

Wigdor, Daniel & Wixon, Dennis 2011. Brave NUI World: Designing natural user interfaces for touch and gesture. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Yatani, Koji & Truong, Khai N. 2009. SemFeel: A User Interface with Semantic Tactile Feedback for Mobile Touch-screen Devices. Proceedings of the 22nd annual ACM symposium on User interface software and technology. 111-120.

Zeng, Jinhua & Sun, Yaoru & Wang, Fang 2012. A natural hand gesture system for intelligent human-computer interaction and medical assistance. Proceedings of the Third Global Congress on Intelligent Systems. 382-385.

Appendix A: Questionnaire

Appendix A(1)

Before the experiment

1. What is your age group?

   A. 10-17 B. 18-24 C. 25-34 D. 35-44 E. 45-54 F. Older than 55

2. Are you left-handed, right-handed, or ambidextrous (use both hands)

   A. Left-handed B. Right-handed C. Ambidextrous

3. What is your experience level in using touchless interaction (which includes the Kinect, WiiMote, or any gesture-enabled devices)?

0 – No experience 1 – Little experience 2 – Casual 3 – experienced 4 – Expert

During the experiment

4. What is your satisfaction level in using the haptic glove?

0 – Very unsatisfied 1 – Unsatisfied 2 – Neutral 3 – Satisfied 4 – Very satisfied

5. What is your satisfaction level when not using the haptic glove?

0 – Very unsatisfied 1 – Unsatisfied 2 – Neutral 3 – Satisfied 4 – Very satisfied

After the experiment

6. Was it easier to control the software with the glove or without it? How so?

7. How easy was it to learn to use this type of interaction (touchless interaction)?

0 – Very difficult 1 – quite difficult 2 – Normal 3 – Quite easy 4 – Very easy

8. What aspects of the interaction, as well as the interface elements, were you most comfortable with?

9. What aspects of the interaction, as well as the interface elements, needs improvement?

10. In your opinion, should haptic feedback be used in touchless interaction? If yes, in which kind of situations do you think haptic feedback is useful?

Appendix B: Task list

Appendix B(1)

You are sitting on your desk, just having finished writing a long report for the course you are taking, User Centered Design (UCD). You saved your report in the directory path “User”. The directory “User” is your home directory, and in this application, it is the root directory, meaning that you cannot go up (back) a level in this directory.

1. Find your report titled “Chapter 2 Summary.txt in your home directory”.
2. You need to move your report to the UCD course folder. Move your report into the following path: “User/Studies/User-Centered Design”.
3. You want to be safe, and create a backup copy of your report. Copy your report, and paste it into “User/Backups”.

Wanting to take a break, you went on the internet and downloaded some cool images that can be used for your desktop background.

4. Go into your Download folder, “Users/Downloads”
5. You realized that you have a few files that you no longer need, and would like to remove some of them. Delete any single text or audio files.
6. Find an image file in there that you just downloaded. There are several images, but the one you are looking for is an image of the Earth seen from space, so you should open each image to find the right one.
7. Now, move this image file into the path “Users/Desktop/Background Images”.

After setting your background image, you realized that in you forgot to copy a song from your computer to your USB flash drive. You connect the USB drive, and it is available in the path “User/F Drive”.

8. Go into “Users/Music/Oldies/” and find the song file “A kiss to build a dream on.ogg”.
9. Play the song, just because you want to listen to it at this moment. You don’t have to listen to the whole song.
10. Then, Copy this song into your USB flash drive.