FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Leap Motion for WIMP based interfaces

Manuel César Bessa Seixas

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Maria Teresa Galvão Dias (Professor)

July 2014

Leap Motion for WIMP based interfaces

Manuel César Bessa Seixas

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: José Manuel de Magalhães Cruz (Professor)

Vogal Externo: Teresa Romão (Professora)

Orientador: Maria Teresa Galvão Dias (Professora)

July 10, 2014

Resumo

A dissertação "Leap Motion para interfaces baseadas em *WIMP*" tem como objetivo fazer uma avaliação do dispositivo Leap Motion no que toca a interação com interfaces *WIMP*.

O Leap Motion é um dispositivo relativamente recente que tem estado a causar algum impacto no mercado devido a forma de interação que este permite com o computador. O aparelho é capaz de detetar a posição, inclinação e orientação dos dedos e da palma de ambas as mãos do ser humano e utilizar os seus movimentos e gestos para controlar o computador. Embora não seja perfeito, o seu preço e precisão têm-lhe permitido conquistar alguma reputação no mercado da interação gestual, antes dominado pela *Kinect* da Microsoft.

O desempenho e precisão do dispositivo vai ser avaliado quando utilizado para interagir com interfaces *WIMP*. Desta forma, poder-se-á identificar se este tipo dispositivo é adequado a este gênero de interfaces e que medidas/alterações serão necessárias tomar nestas, caso necessário ou possível, para melhorar o seu desempenho.

A tarefa avaliada foi a de seleção. Para efetuar esta avaliação serão calculadas sete medidas de precisão através das quais será possível classificar o movimento efetuado pelo ponteiro quando controlado pelo dispositivo de forma extrair conclusões. Isto poderá permitir distinguir qual o dispositivo mais adequado para cada ação ou tarefa quando os seus desempenhos são semelhantes. Para complementar a avaliação será também calculado o seu desempenho e taxa de erro. O rato de computador e painel táctil foram também avaliados para servir de comparação.

Para converter a posição e movimento de um dedo, escolhido pelo utilizador, em ponteiro foi utilizada a aplicação comercial *Touchless*, exclusiva ao aparelho.

Face os resultados obtidos, o desempenho do Leap Motion mostrou-se inferior aos restantes dispositivos, quando utilizado neste tipo de interfaces. De forma a tentar perceber se este resultado é devido ao dispositivo em si e não ao gesto utilizado, complementou-se o estudo realizando a uma nova experiência, semelhante a anterior, utilizando um gesto novo desenvolvido no âmbito desta dissertação.

No fim, mesmo apesar de o novo gesto trazer um possível resultado positivo a nível de desempenho e conforto, os utilizadores continuaram a preferir o rato de computador comum. Como tal, mesmo apesar de o dispositivo ser compatível com este tipo de interfaces, é pouco provável a sua adoção pelos utilizadores como uma alternativa viável de dispositivo de entrada para controlar o seu computador e interagir com interfaces baseadas em *WIMP*.

Abstract

The dissertation "Leap Motion for WIMP based interfaces" aims to perform an evaluation on the Leap Motion device, namely when used to interact with WIMP interfaces.

The Leap Motion is a relatively recent device that has been causing some impact in the market due to the way the interaction with the computer is performed. The device is able to detect the position, inclination and orientation of the human palm and fingers of both hands and use their movements and gestures to control the computer. While not perfect, its price and precision have permitted the device to win some reputation in the market of gestural interaction.

The device throughput (performance) and precision were evaluated when used to interact with the WIMP interfaces. This way, it's possible to identify if this kind of device is adequate to this type of interfaces and what measures/changes are needed, if necessary, to improve the Leap Motion performance.

The task evaluated was a "selection" task. To perform this evaluation seven accuracy measures were calculated. Through these it will be possible to classify the movement performed by the pointer when controlled by the device and extract conclusions. This may allow to differentiate which device is better suited for each action or task when their performance are similar. To complement this evaluation, the device throughput and error rate were calculated. The computer mouse and the touchpad were also evaluated to serve as a comparison,

To convert the position and movement of a finger, chosen by the user, to a pointer a commercial application named Touchless, exclusive to the device, was used.

Taking in account the attained results, the Leap Motion performance was lower than the remaining devices when used in this kind of interface. In order to try to understand if this result was really due to the device and not to the used gesture, the study was further complemented by carrying out a new evaluation, similar to the last one, but using a new gesture developed within this dissertation.

In the end, even though the new gesture may have brought an improvement to the interaction in terms of performance and comfort, users still preferred the computer mouse. As such, even though the Leap Motion is compatible with this type of interfaces, it is unlikely the users will choose it as a viable alternative to control their computers and interact with WIMP interfaces.

Acknowledgments

First and foremost, I would like to thank (a lot) the persons who participated in the experiments. Thank you very much for all you effort (and suffering)! Without those, there would be no results nor feedback in this document.

I also would like to thank all CITAR members, for their help, suggestions and company during this project. A special thanks to Eng. P. for all the cookies, cakes, Russian cuisine, patisseries, conversation, soap operas and discussions about tsunamis!

Finally, at the end are always the best, I would like to thank Professor Jorge C. S. Cardoso for the analyses script and for babysitting me the whole semester. I would also like to thank as much Professor Teresa Galvão for accepting being my supervisor and for all the support, knowledge, patience and detection of errors (which were not few) that no doubt contributed to make this document a better one.

Manuel César

Contents

List of Figures

List of Tables

Abbreviations

3D	Three Dimensional
API	Application Programming Interface
GUI	Graphical User Interface
ID	Index of Difficulty
IDe	Effective Index of Difficulty
IDE	Integrated Development Environment
ISO	International Organization for Standardization
HCI	Human – Computer Interaction
JRE	Java Runtime Environment
MDC	Movement Direction Change[1]
ME	Movement Error[1]
MO	Movement Offset[1]
MV	Movement Variability[1]
ODC	Orthogonal Direction Change[1]
OS	Operating System
TAC	Task Axis Crossing [1]
TRE	Target Re-entry[1]
SDK	Software Development Kit
WIMP	Windows, Icons, Menus and Pointers

[1]These terms were coined in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)] by the authors.

Chapter 1

Introduction

1.1 Context

The Leap Motion controller [(Leap Motion n.d.)], developed by an enterprise with the same name, is a device that appeared recently on the market and is revolutionizing the way people interact with their computer. This controller is able to recognize human hands, allowing "communication" between man and machine to be done through motions and gestures.

Note that this way of interaction is in no way a novelty, human computer interaction using these means is long being used, of which are examples Microsoft's *Kinect* [(Microsoft Corporation 2013)] and Nintendo's *Wii* home console controller [(Nintendo of America, Inc. n.d.)]. Nonetheless, the Leap Motion controller is generating a hype on the market with its promises and functionalities. However, no matter how promising, there are yet to be studies proving its efficiency and usability.

The Leap Motion controller already provides the possibility of being used as a pointing device. In other words, the user can utilize it as a way to control the pointer present on the desktop/laptop's screen as he/she did with a common computer mouse. However, is this useful? How good is the Leap Motion when compared to the controllers we use nowadays? And even if better, how much would we have to change what already exists to use this new technology? Will our interfaces require a drastic change or an adaptation would be possible without significant effort? How much would the WIMP based interfaces have to change? Is there even a need for these changes?

The WIMP (meaning Windows, Icons, Menus and Pointers) is a kind of interface design, where the user interacts with a computer through the elements the acronym is composed of. One example of this interface would be the Microsoft Windows' Desktop, as the reader can observe in **Figure 1**.

Figure 1: An example of a WIMP based interface

All of these questions are yet without an answer, not because the tasks required to get an answer are difficult, but due to the recent release of the Leap Motion controller. The objective of this dissertation is to try to answer some (if not all) of those questions.

This kind of study is inserted in the Human – Computer Interaction (HCI) thematic and will be done at Center of Academic Research of School of the Arts (CITAR), at the Portuguese Catholic University, in cooperation with *Faculdade de Engenharia da Universidade do Porto* (FEUP). The dissertation will be supervised by Professor Teresa Galvão, from FEUP, and Jorge Cardoso, from CITAR.

1.2 Motivation and Objectives

Being this a new device, still with an unknown number of capabilities and with a (more or less) new way for users to control their long-time used machines in a more intuitive manner, it would prove fruitful to study and evaluate the Leap Motion as a pointing device. If truly better than the common computer mouse, future generations or first time computer-users could start to use it in a day-to-day basis to work, play and have fun with their machines. All this in a more humanly fashion.

To questions like “How do I select this?” or “How do I grab that and carry it here?” no answers like “You must move the mouse like this” or “You need to select it with the left mouse button and, without releasing it, drag it to the place you want“ should be given. Instead, the answer would be “like you do it in real life”, “point at it” or “pinch it and drag it to the place you want”, an answer more easily understandable by us, humans.

It’s like all that magic and fantasy we have seen on the cinema or read in books suddenly springed to life. It is magic made real. And is there anything more wonderful than magic?

1.3 Methodology

In order to discover if this is possible, the following steps had to be taken:

1. Study the Leap Motion Software Development Kit (SDK)

This step is required to understand what the Leap Motion is already able to do and how it does it. Without knowing its capabilities it is not possible to know what functionalities to evaluate. For example, to evaluate the device in a selecting task, one must know what kind of movements/gestures are able to perform the said action.

Also, by studying it, it will be possible to determine how to add new functionalities such as, for example, new gestures that would improve the experience and the Leap Motion usability.

2. Evaluate the Leap Motion controller

To do this a similar experiment to [(MacKenzie, Kauppinen and Silfverberg 2001)] will be performed. In the paper, the authors propose a set of new measures to analyze the computer's pointer movement performed during each trial. With this, even if two devices seem close in Throughput (read performance), it is still possible to determine which device is better suited to a certain task. With it, specific values, namely Target-Re-entry, Task Axis Crossing, Movement Direction Change, Orthogonal Direction Change, Movement Variability, Movement Error and Movement Offset, will be calculated.

This is somewhat an innovative measure, since this information is not usually determined by many authors, and the extra data could prove insightful to determine in which circumstances the Leap Motion is better than other devices.

3. Complement the evaluation with ISO 9241-9:2000 and Error Rate

To complement the evaluation of the device as a pointing device, the International Organization for Standardization (ISO) 9241-9:2000[1] standard [(International Organization for Standardization 2000)] will be used. From it, the necessary steps to calculate the device performance (or Throughput, as named by the standard) will be extracted. While not part of it, Error rate will also be used.

Both these metrics have been long known and frequently used in many other works [(Cheong, et al. 2011), (Douglas, Kirkpatrick and MacKenzie 1999), (Natapov, Castellucci and MacKenzie 2009), (MacKenzie and Jusoh 2001), (McArthur, Castellucci and MacKenzie 2009), (Natapov and MacKenzie 2010)].

[1] The ISO full name is "Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 9: Requirements for non-keyboard input devices"

More information on these metrics will be presented in section **2.2.2** and section **2.3**.

4. Evaluate the computer mouse and the touchpad

In order to perform a comparison between the Leap Motion and other devices, the same evaluation should be performed on both. The devices chosen to do this comparison were the computer mouse and the Touchpad. While there are already several evaluations of these tools, they were probably performed on a different software and hardware. While the tests are the same, the way they are coded, implemented and performed may be different; the performance of the software and hardware they are ran upon may also be different. This will surely give rise to differences in the data acquired. As such, an evaluation of the all the peripherals should be done under the same circumstances to assure that results have the same basis.

5. Compare the Leap Motion with. Mouse and Touchpad.

After the evaluation is performed, the data gathered from each apparatus must be compared in other to determine what the Leap Motion does best (or worse) and what it does different than the other devices. Based on this, an answer to the questions placed above should appear.

6. Based on the results, discover if changing the gesture would improve the review of the Leap Motion by the users.

After performing the initial experiment, the conclusion reached was that the Leap Motion was not well liked by the users who experimented with it, being its adoption to these tasks unlikely to happen. However, it was not possible to determine if this result was due to the device or to the gesture being used. As such, another experiment was conducted using another gesture for executing the same tasks.

7. Present the results and propose solutions for interaction with WIMP based interfaces using the Leap Motion.

These solutions (if any) may came in form of a guideline, a collection of suggestions that, if followed, will lead to a better use, from usability point of view, of the Leap Motion in the WIMP based interfaces.

Also, if the gesture developed and used during the second evaluation proves to be useful to the users performing the experiment, a new gesture could be proposed as an extra way of controlling and using the Leap Motion.

1.4 Structure of the Dissertation

The following chapters of this dissertation will deal in more detail with themes discussed in this introductory chapter.

Chapter 2 presents more information about the Leap Motion, namely where it is used or where it has been studied. In other words, a State-of-art related to the Leap Motion. There will also be an explanation of the pointing device's evaluation techniques found, including the one that will be used to evaluate the Leap Motion in this dissertation.

Chapter 3 deals with the implementation of the application required to perform the evaluation, specifying the technologies used, architecture, graphical user interface and developed gestures, among other nuances.

Chapter 4, on the other hand, will deal with the flow and schedule of both experiments performed in this dissertation, detailing the equipment necessary to it, the process and information of the gathered participants.

Chapter 5 presents all the data gathered from the experiments and the conclusions attained from the assessment. Several solutions, namely a collection of guidelines, are also offered.

Finally, Chapter 6 will be the concluding chapter, also including possible future work.

Chapter 2

Related Work

In this chapter the Leap Motion device will be explained in more detail, from its composition and capabilities to other technical details. The amount of information that was made available to public by its creators is somewhat restricted, since the device is still under a patent which was filed very recently. As such, the information presented will be either extracted from the promotional campaign made by "Leap Motion", the enterprise responsible for the creation of the device, reviews done in magazines, journals or other respectable press and in scientific studies/articles that rely or focus in the study of the equipment.

The number of studies with Leap Motion is still quite small at the moment since the device was only made available recently.

2.1 The Leap Motion Controller

The "Leap Motion controller" [(Leap Motion n.d.)], as seen in **Error! Reference source not found.**, is a small device, 7,6cm x 3cm x 1,3cm (length x width x height),created by the company with the homonymous name, "Leap Motion", that lets users control the computer and manipulate objects onscreen through gestures and movements performed with their hands.

Figure 2: The Leap Motion Controller (Source: (Leap Motion n.d.))

To operate the device there is only need to plug it in a computer, either Microsoft Windows or Mac OS, through an USB cable, included in the package, and install the software required for it to be recognized. This software can be downloaded from the Leap Motion main website[1].

For development, the enterprise has already made available the Leap Motion SDK and the application programming interface (API). These are available to various programming languages, namely *C++*, *Objective-C*, *C#* (for *Unity*), *Java*, *Python* and *JavaScript*. These can be download exclusively at the Leap Developers Portal[2].

The equipment also has an online store[3], named “Airspace”, from which users might buy applications to use with their device. Some are free, some are restricted to a specific OS. These applications have wide range of functions and several examples of them will be given in **2.1.1**Applications or projects using Leap Motion.

The Leap Motion controller was being sold for $79.99 (US dollars), €89.99 + import taxes for Europe, which is considered by many magazines a very cheap alternative to the Microsoft’s *Kinect*, to whom the motion sensing market used to belong.

Due to its small size, the device is being integrated into laptops, step being taken by Leap Motion (enterprise) in partnership with Hewlett-Packard (also known as HP) and Asus, embedding the controller directly on the computer, as stated in [(Pogue 2013)].

To perform its functions, the equipment creates a 3D interaction space above it, 8 cubic feet (approximately 2.43 meters) according to [(Garber 2012)], being capable of sensing anything that goes through it and converting it to data. That data is then processed, structured and made available to any application through the API or library of the device. In its promotional campaign, chopsticks and a pencil were used to control the device [(Leap Motion 2012)]. More importantly, the device is also able to follow and track each one of our 10 fingers (as long as they don't overlap or stand too close), translating the users’ gestures and movement to a respective action (this action varies from application to application) with none or very little latency.

In terms of precision, the Leap Motion is being marketed as an equipment with 200 times the precision of Microsoft’s *Kinect*, being able to accurately detect a hundredth of a millimeter (this, however, does not hold true in a real environment, as proved by [(Weichert, et al. 2013)]) and this level of sensitivity can be customized, according to [(Garber 2012)]. [(Markowitz, Wave Your Hand, Control Your Computer. 2013)] adds that the Leap Motion is capable of capturing movements up to 290 frames per second, which is higher than Microsoft’s *Kinect* 30 frames per second. However, this is not true at all times. During the development of this dissertation it was verified that the number of frames per second varied depending on: the environment in which the Leap Motion was being used, if it was able to detect something in its detection range or depending on its tracking priority (named level of sensitivity by [(Garber 2012)]). There are three possible

[1]Leap Motion setup webpage: https://www.leapmotion.com/setup (accessed June 16, 2014)
[2]Leap Motion Developer's page: https://developer.leapmotion.com/ (accessed June 16, 2014)
[3]Airspace Store: https://airspace.leapmotion.com/ (accessed June 16, 2014)

tracking priorities: "Precision", less frames but more accuracy, "High Speed", more frames but less accuracy, and "Balanced", a middle ground between the last two options.

The process to achieve these results, however, is a secret. In [(Gardiner 2012)], the author states that the Leap Motion “uses a combination of infrared LEDs and 1.3-megapixel camera sensors to monitor movement in an 8-cubic-footfield” and “Leap’s software runs custom algorithms to convert what the device sees into a 3-D map of the user’s hands”. Again, [(Markowitz, Wave Your Hand, Control Your Computer. 2013)] goes a little further stating that “The cameras have a 150-degree field of view that allows them to track your finger movements in a 2-foot-wide[1] area”. However, this information is not definitive as the process to make so happen is still shrouded in mystery and won’t be revealed anytime soon due to the patent on the Leap Motion Controller.

The controller already has some gestures implemented in itself, allowing the application or program to query directly for them instead of analyzing all the captured data. There are four recognized gestures, namely the "Circle Gesture"[2], "Key Tap Gesture"[3], "Screen Tap Gesture"[4] and "Swipe Gesture"[5].

Nevertheless, it is also possible to create custom-made gestures / movements and give them specific actions. For example, according to [(Marks 2012)], the analyzed application already had motions (or gestures) for clicking, grabbing, scrolling and "zooming in", being a pinching motion used to zooming in on an image.

.

While a powerful tool, the Leap Motion has received mixed reviews. Some newspaper and magazines describe it as the next successor to the computer mouse while others indicate that the device is only good for gaming and design, unable to replace the mouse. From a usability point-of-view, in [(Savage 2013)], many persons who experimented with the device complained that different applications used different gestures to perform similar functions. In other words, the controls learnt availed to nothing since these motions might do exactly the contrary in another application. The author also adds that the interfaces we use nowadays, like menus, might not be as useful with the Leap Motion as they were with a Keyboard or mouse. [(Pogue 2013)] does a particular nasty evaluation of it, showing most of the device’s “rotten” aspects. It also has been showed, in [(E. S. Silva, J. A. Abreu, et al. 2013)], that when fingers overlap (for example when

[1] 2 foot is equal to 609.60 millimeters or 0.60960 meters. (Source: http://www.metric-conversions.org/length/feet-to-meters.htm?val=2, (accessed May 29, 2014)

[2] For more information: https://developer.leapmotion.com/documentation/java/api/Leap.CircleGesture.html (accessed May 29, 2014).

[3] For more information: https://developer.leapmotion.com/documentation/java/api/Leap.KeyTapGesture.html (accessed May 29, 2014).

[4] For more information: https://developer.leapmotion.com/documentation/java/api/Leap.ScreenTapGesture.html (accessed May 29, 2014).

[5] For more information: https://developer.leapmotion.com/documentation/java/api/Leap.SwipeGesture.html (accessed May 29, 2014).

the user crosses fingers) the controller is unable to distinguish between the two, counting and tracking them as only one.

Finally, the Leap Motion device is scheduled to suffer an update on its API. Version 2.0 is now (29 of May of 2014) on public Developer beta, with the promise of being released commercially to the public by the end of the year (end of 2014), but without a specific date.

This new version introduces several upgrades. As stated in [(Leap Motion 2014)] this improvement will set up a new skeletal tracking model with extra information about the hands and fingers. This will allow to track all the fingers and hands even if they are partially occluded through the internal hand model, to know if a pinch (two fingers touching each other) or a grab (hand curled into a fist) is happening, which hand is the left and right without extra calculations, to get finger information by their names and to know the position and rotation of each finger's bone.

This upgrade, however, may rend some applications and programs useless, since the information returned from the device may now be completely different than the expected. One example of this are gestures and motions that were based in the number of fingers being shown. These will no longer work due to internal hand model. As long as one hand is detected, this model can be generated and the five fingers' position can be extrapolated, resulting in the number detected fingers always being five. If an action happened when the number of fingers shown was three, then this action will never happen. These applications will have to be updated or, if the process no longer works, find another solution to achieve the same results they did before.

The version 2.0 of the API was tested during this dissertation, however, due to being in its early stages (it seemed somewhat buggy and didn't brought noticeable improvements) it was not used during the development of this work. The commercial, version 1.0, API and libraries were used instead.

2.1.1 Applications or projects using Leap Motion

In this section the reader will find several examples of applications already developed for the Leap Motion controller. When this dissertation was being written the number of applications already went above 150 with the promise of more and more coming to the market. These are composed of original ones, made exclusively for the Leap Motion and taking full advantage of the its capabilities, as well as adapted ones already famous in other OSes (Operating Systems) ported to work with the device.

The Leap Motion device is mainly applied to the multimedia field, for entertainment purposes, although there are already several applications of the device with a more serious take.

Related Work

Some of the presented examples will be accompanied by a link containing a video, which the reader might watch to get a better grasp of the application function and objective.

1. Creating Music with the Leap Motion

The Leap Motion has already been applied to the music field. One good example of that is the program GECO[1], a program that acts as the middleman between the user and other music related programs that use protocols/frameworks related to MIDI, OSC or *CopperLan*. The application can be controlled through the hands movement and gestures, being these capture by the device, which are then translated into the respective action[2].

This way the DJ may place special effects and other musical elements on the playing music (it's not possible to produce or manipulate sound directly) simply placing is hands above the device and move them up-and-down or left-and-right. The action performed by each gesture can also be personalized in order to fit the user[3].

Unfortunately, another input device is required on some steps of the application (for example, the personalization process), making the Leap Motion a complement and not exactly a complete substitute of other input devices.

2. Autodesk and Leap Motion on the commercial world

Autodesk[4], a famous enterprise dedicated to software and services related to 3D design, engineering, and entertainment, has already integrated the Leap Motion in the modeling and animation process of Maya[5], a 3D animation software.

The Leap Motion can be used as complement to the computer mouse or touchpad, allowing to model digital objects in a similar fashion to modeling clay, grabbing and dragging digital objects in three dimension environment without having to constantly switch camera views, facilitate the animation process and perform certain simple actions that were performed using other input devices[6].

---

[1] Application Webpage: http://uwyn.com/geco/ (accessed May 30, 2014)
[2] GECO promotional video: http://www.youtube.com/watch?v=N8uFiJZpkgY (accessed May 30, 2014)
[3] A more detailed view of the application: http://www.youtube.com/watch?v=RCwiwYAVxOY (accessed May 30, 2014)
[4] Autodesk main website: http://www.autodesk.com/ (accessed May 30, 2014)
[5] Maya webpage: http://www.autodesk.com/products/autodesk-maya/overview (accessed May 30, 2014)
[6] Some examples: https://www.youtube.com/watch?v=qjSWTpVlvLI (accessed May 30, 2014)

3. Leap Motion as an entertainment tool

During October 3, 2013, Autodesk organized a monthly event, named Autodesk Design Night, and one of the proposed challenges was for the participants to build a "Jack-o'-Lantern" (the carved pumpkin face often seen in the American Halloween festival) mascot, using the Leap Motion device and Maya.

The pumpkin and body elements (helmets, glasses and other accessories) were already modeled, having the user only to select and drag the elements he/she liked to build his/her mascot[1].

In here, the Leap Motion was capable of substituting the computer mouse in the pointing, selecting and dragging tasks.

4. Leap Motion as interactive tool on (immersive) virtual environments

SpaceX[2] is enterprise dedicated to design, manufacturing and launching of rockets and spacecraft in order to enable people to live in outer space. Recently, the company also experimented with the Leap Motion device[3], applying it in the modeling, multimedia and virtual reality.

With the help of the device it's already possible to manipulate and interact with a single virtual object present either on a computer screen, projections on glass, 3D projection (with the help of 3D glasses) and immersive virtual reality. Tasks like zooming in and out, rotating, translating and spinning in a continuous motion are achievable without the help of a mouse or keyboard.

5. Leap Motion as modeling tool

Freeform[4] is an exclusive modeling application for the Leap Motion device, available for free in Airspace (Leap Motion application store). With it, the user can use his/her fingers to model an object in materials resembling glass, clay, plastic or porcelain, select a different tools and brushes to sculpt and paint with then, directly edit the wireframe and even change the background environment[5].

---

[1] An excerpt of the Autodesk Design Night: http://www.youtube.com/watch?v=SYlgQB6Avy8 (accessed May 30, 2014)

[2] SpaceX main website: http://www.spacex.com/about (accessed June 2, 2014)

[3] Here is one example: http://www.youtube.com/watch?v=xNqs_S-zEBY&list=FLyChSY1QQ1pID7o68CSJmvg (accessed June 2, 2014)

[4] Application page on Airspace: https://airspace.leapmotion.com/apps/freeform/windows (accessed June 2, 2014)

[5] Using Freeform: https://www.youtube.com/watch?v=rod-Gg3j_U0 (accessed June 2, 2014)

While seemingly a simple application, thus somewhat limited to what other tools in the same field might offer, like Autodesk, it’s also the first example where the Leap Motion is exclusively used for the task at hand, without requiring the help of a mouse or other input device.

6. Leap Motion for Photoshop - Ethereal

Ethereal[1] is an Add-on for Adobe Photoshop[2], an image editing software, which allows the operator to use most of the program functionalities using his/her fingers and hands by using the Leap Motion.

Through the movements of hands and fingers it is now possible to draw lines, choose their thickness (based on the distance between a perpendicular invisible plane to Leap Motion and one finger), move an image, define the brush size (diameter of the circle drawn) and which color to paint[3].

The add-on also accepts the use of objects, as long as it has the necessary length and thickness to be detected by the Leap Motion, like a pencil for example, instead of a finger to perform the same tasks. This way the process becomes more intuitive since the user can now draw just as he/she is used to in real life, sans friction.

There is, however, a little catch. Not all the application’s functionalities are available, since some are incompatible with the device, requiring another input device, like the mouse, to work.

7. Leap Motion with common applications – Google Earth

Some common, every day applications are already compatible with the device. One example is Google Earth[4], a free software that allows the user to explore the world through satellite photos of the planet Earth, is now ported to work with the Leap Motion, adding an interactive and funny way to “travel around the world”.

The user utilizes his/her hand's movement to interact with the application. The hand is like an airplane, soaring through the earth. As the hand changes direction so does the content onscreen and moving the hand closer or away from the device allows to zoom in and out of the planet[5].

In here, the Leap Motion is completely capable of replacing other input devices, such as the Mouse or Touchpad, having the same functionalities as those.

[1]Add-on page on Airspace: https://airspace.leapmotion.com/links/ethereal (accessed June 2, 2014)
[2]Adobe Photoshop main page: http://www.adobe.com/pt/products/photoshop.html (accessed June 2, 2014)
[3]Drawing a picture with Ethereal: https://www.youtube.com/watch?v=joIw5MuC1aM (accessed June 2, 2014)
[4]Google Earth Webpage: http://www.google.com/earth/ (accessed June 2, 2014)
[5]Soaring through Google Earth: http://www.youtube.com/watch?v=RebX7YEn3GQ&list=PLZgjuTxMC0h1k9pBKUNuDCmEF0aWHLYV (accessed June 2, 2014)

8. Customizing Leap Motion gestures to a OS and Web Browsers

BetterTouchTool[1] (version 0.970) is an application, currently in beta, that allows the user to configure motions for many Apple's devices (like mouse, touchpad and remote), normal mouse, keyboard and for the Leap Motion as well. These gestures can be mapped to common actions done on the Mac OS, which uses a WIMP interface, allowing the user to customize what movements do what.

Focusing on the Leap Motion component, the software offers twenty-seven pre-defined gestures (or offered since more may have become available since June 2014) and the development team promises to keep adding more.

This application should be a strong inspiration (or competition) on the gestures to develop and implement for the evaluation to be performed in this dissertation.

Similar applications to the last one exists for web browsers. One example of this is OSControl, available for the Internet Explorer[2] and Google Chrome[3], although only for Windows operative system.

Once again, the action performed by each gestures can also be personalized by the user.

This software also allows the pointer to be controlled through the Leap Motion, however, as stated by the author, its objective is to complement the mouse and keyboard and not to replace them.

9. Leap Motion and Video Games

The Leap Motion has also been applied as a controller for video games. One example of this is "Fruit Ninja"[4], a renowned game for the Smartphone devices, now available in Airspace[5] to be played with the Leap Motion. The objective is to slice the appearing fruit before it reaches the bottom of the screen while avoiding the arbitrary bombs.

In the original version, the user used his/her finger to slash across the fruit. With the Leap Motion the same happens but instead of rubbing the touchscreen the user places his/her hand above the device and performs the slashes by moving a finger[6].

---

[1] Application Airspace page and demo video: https://airspace.leapmotion.com/apps/bettertouchtool/osx (accessed June 2, 2014)
[2] OSControl - Internet Explorer Edition: https://airspace.leapmotion.com/apps/oscontrol-internet-explorer-edition/windows (accessed June 2, 2014)
[3] OSControl - Google Chrome Edition: https://airspace.leapmotion.com/apps/oscontrol-chrome-edition/windows (accessed June 2, 2014)
[4] Fruit Ninja demo video: http://www.youtube.com/watch?v=UJlNFA4ioPQ (accessed June 2, 2014)
[5] Fruit Ninja on Airspace: https://airspace.leapmotion.com/apps/fruit-ninja/windows (accessed June 3, 2014)
[6] Fruit Ninja being played with the Leap Motion: http://www.youtube.com/watch?v=ynN9oFaXWgU (accessed June 3, 2014)

While simple, this is one example where the Leap Motion can replace a touchscreen, the most used input device to control smartphones.

10. Leap Motion on Education

The Leap Motion has also been applied to the education field, namely on Biology. Children (Grow-ups as well!) can now learn the names and position of the bones that compose the human skull or even dissect an animal (namely a Tarantula, Rhinoceros Beetle, Caterpillar, Butterfly, Earthworm or a Starfish) by using their hands and fingers.

With "Cyber Science - Motion"[1] the user can use his hand to zoom in and out of the human skull in order to see each bone's details and position, point with the index finger at a bone to reveal its name and move it from its original position by pointing at it with the index finger and showing the thumb[2]. Several challenges are also presented (for example, bones start out of their original positions and the user must place them here they belong) so the user might test his/her knowledge and have fun. With "Cyber Science - Motion: Zoology" the concept remains the same, only this time the user explores the animal anatomy and its particularities.

Finally, the application can resort to the mouse or keyboard to perform certain tasks, which makes the Leap Motion the primary tool but not an exclusive one.

11. Applications adapted to work with the Leap Motion

Several applications, from which are example *Stan Lee's Verticus* and *Allthecooks Recipes* (Fruit Ninja, which was presented a few paragraphs before, in section **Leap Motion and Video Games**, is also an example), have already been adapted to work with the Leap Motion controller.

Stan Lee's Verticus[3] is a video-game application created for iOS, now also playable at the desktop thanks to the device. In it, the player controls a superhero falling to the center of the Earth while avoiding obstacles by using his/her hands. To make the hero gain momentum the player puts his hand parallel to the device and to decrease its speed the player puts his hand vertical to the device (as if making a stop sign aimed at the computer screen). The hero's position onscreen is exactly the same as the player's hand position in relation to the Leap Motion.

Allthecooks Recipe[4]*s* is a social cooking application available for iOS, Android, Windows Phone. Now it can also be used in a desktop computer thanks to the Leap Motion. The objective

[1] Cyber Science - Motion Airspace page: https://airspace.leapmotion.com/apps/cyber-science-motion/windows (accessed June 4, 2014)

[2] Cyber Science - Motion demo: http://www.youtube.com/watch?v=ekIMK7ORY_M (accessed June 4, 2014)

[3] *Stan Lee's Verticus* Airspace page and demo: https://airspace.leapmotion.com/apps/stan-lee-s-verticus/windows (accessed June 6, 2014)

[4] *Allthecooks Recipes* Airspace page: https://airspace.leapmotion.com/apps/allthecooks-recipes/windows (accessed June 6, 2014)

of the application is to stimulate people to cook and eat at home and also to promote creativity and sharing of cooking recipes through the members of the community. To do this, the application allows the user to create and manage a profile, upload his/her recipes and browse over all the available recipes made by other people. Using the Leap Motion, the user can browse through them by showing all fingers of one hand and swiping his/her hand to the left or right (the recipes are all presented in a horizontal line). To select a recipe, the user points a finger at it and hovers it over the magnifying glass that appears. To browse the recipe the user shows all his/her fingers from one hand and swipes it in the direction he/she wishes to scroll the page. When finished, the user can return to the last screen by pointing at the arrow sign in the middle left of the page and drag his/her finger down[1].

The meaning behind this section is not that the Leap Motion is better or a substitute to the mouse, touchpad or touchscreen but that developers are betting on the device, promoting it as reliable and powerful tool. With this, even if the controller is enable to adapt to the WIMP interfaces, there is it still a chance for it to remain on the market and a find a target (mayhap niche) audience to support it. If the device were to disappear then all the results here contained would be for naught.

12. Leap Motion applied to Medicine

The Leap Motion device has also been applied to less ludic areas like Medicine, in order to try to decrease waiting time and increase performance.

One of the examples is a medical procedure. The medical staff needs to consult certain medical files (X-rays, for example) that are sometimes essential to the operation in course. When using a mouse to browse through these, the surgeon had two choices: remove his/her gloves to operate the device, consult the information, replace the gloves when the task was over and return to the operation or ask another member of the team to perform these actions instead (this, however, would require coordination between both). In other words, the existing solutions were either too time-consuming or low-performance.

With the help of the Leap Motion, the surgeon can perform the browsing directly with his/her equipment on and without requesting help from another staff, hastening the process[2]. This makes the Leap Motion an excellent controller to have on situation like these since it can overcome the computer mouse’s utility.

[1] *Allthecooks Recipes* quick demo: https://www.youtube.com/watch?v=31-qV32fYfM (accessed June 6, 2014)
[2] An example (by TedCas and Leap Motion): http://www.youtube.com/watch?feature=player_embedded&v=6d_Kvl79v6E (accessed June 7, 2014)

13. Are there more?

Yes, many more. As stated before, the number of applications available for the Leap Motion is still growing. However, the number of examples presented should be enough to portray all the types of applications available.

The following section will leave the commercial side of the device, focusing on a number of experiences, more scientific in nature, performed with the controller.

2.1.2 Experiments with Leap Motion

Since the Leap Motion controller was only recently released (22nd of July, 2013, as stated in [(Leap Motion, Inc. 2013)]), the number of experiments with the device is still low. Nevertheless, some have already been performed and will be described in this section.

1. Studying the Leap Motion precision and repeatability

In [(Weichert, et al. 2013)] the authors studied the precision and repeatability of the Leap Motion in order to determine if the marketed precision, “hundredth of a millimeter”, is also possible in real environments. In order to do so, an experiment was performed with the assistance of a robotic arm. A preliminary version (a different version from the current commercial available) of the Leap Motion was used.

According to the authors, accuracy is “the ability of a 3D sensor to determine a desired position in 3D space” while repeatability is the capacity of the “sensor to locate the same position in every measurement repetition”.

The process to determine both was as follows: The robotic arm hold a pen in its hand and was programmed to place the tip in several real world known positions. These positions would then be compared to the ones acquired by the controller, being the difference between each other the precision. These measures were repeated several times in order to find repeatability.

However, before the experiment was performed there was the need to establish a relation between the coordinate system of the robot’s pen position (real world coordinates) and the device internal coordinates. This step was necessary because the internal coordinates of the device didn't match the real world.

Furthermore, two types of accuracy were calculated, static and dynamic. In the static, the tip of the pen was placed in a position and remained still until its location was captured. In the dynamic, the robot was programmed to describe a pre-determined path, being the path drawn the same as a sinus function.

Pens with different diameters were also tested in order to determine if the size would alter the obtained results.

Based on the results, the authors concluded that, in the static experiment, the accuracy obtained was less than 0.02mm (millimeters). It was also shown that the diameter of the pen didn't affect the result. In the dynamic experience, the accuracy obtained was less than 2.5mm. Repeatability was 0.17mm, on average.

The authors concluded that in real environments it's impossible to have the marketed accuracy but, when compared to other gestured based devices, the device remains the most precise.

2. Usability testing on travel techniques using the Leap Motion as a form bare-hand interaction.

In [(Nabiyouni, Laha e Bowman s.d.)] the authors perform a usability testing in order to find which of the travel techniques is the most efficient in bare-hand interaction. Five techniques were tested in a set of 3 tasks and the interaction was performed through the use of the Leap Motion controller.

A 3D virtual scenario, modeled as a city, was used to perform the tests. The objective was to control the camera in order to perform a set of task, controlling its direction and speed through the developed techniques. The test can be likened to a helicopter traveling over a city, being the tasks the objectives that the helicopter must accomplish and the techniques the different manners the driver has to control the vehicle.

The techniques developed were based on a "Camera-in-hand" metaphor, where the Leap Motion workspace was directly mapped to the virtual world, and an "Airplane" metaphor, that, similar to driving a vehicle, had the camera always moving straightforward being the user responsible for controlling its velocity and orientation (the orientation was the same as the hand). The "Airplane" metaphor also originated four different techniques to control speed, either with one or two hands. There was also the possibility that speed could be changed either continuously (change occurred smoothly) or discretely (the speed would change suddenly).

The tasks to perform were "Absolute Travel", where the user had to travel from a starting point to a known ending point, "Naïve Search", where the user had to find a number of obstacles (12 obstacles total) in the minimum possible time (120 seconds was the time limit) and "Path Following", where the participants must try to remain within a set path, travelling through the middle of a number of circles.

Based on the results from the user evaluation, the authors concluded that a continuous speed control could provide a better user experience.

The authors praise the Leap Motion accuracy yet found several defects while performing the experience, being those the inability to distinguish between fingers when they were close or crossed over and the loss of tracking capabilities when the hand is rotated, rolled or pitched, over eighty degrees.

3. Corel Painter Freestyle with Leap Motion Controller, painting in the air.

This experience, contrary to others, has a more commercial tone. In the poster [(Sutton 2013)] the author applies the Leap Motion Controller to an existing application, the Corel Painter, to do something similar to some of the examples presented before (see **Autodesk and Leap Motion on the commercial world** or **Leap Motion for Photoshop - Ethereal**, section **2.1.1**), be a possible substitute of the computer mouse. The result was a similar commercial application but with a new form of interaction.

In [(Malcolm 2013)] the artist performs a small demonstration of the resulting application.

4. Writing in the air. Recognizing gestures and movements as letters using the Leap Motion.

In [(Vikram, Li and Russell 2013)], the authors use an existing algorithm and improve it so it can detect and recognize words and letters written in the air, in real time, with the assistance of the Leap Motion Controller. This detection had also to be accomplished without a gesture indicating the end of the character or word, a commonly used step to signal the software that the user had finished writing, making this experiment an innovative one.

The Leap Motion was responsible for capturing the position of the input over time. The captured sequence of points (a line is a sequence of points, a character is one or several lines combined) would then have to be compared with a dataset to determine which letter, or letters, had been written.

However, before the comparison was possible, the dataset had to be built. To do this, several samples had to be collected. This task involved several users writing in their own way and style, both lowercase and uppercase, in order to gather an extensive number of examples. Furthermore, this comparison had to be performed extremely fast (this would be achieved through optimization) and be able to solve all the conflicts (for example, one letter being confused with another) that showed up.

According to the authors, the developed application was able to perform as expected with the promise of expanding its functions in a future work. The Leap Motion controller was up to the task, capturing the input with necessary precision to be recognized.

In the work to be developed in this dissertation, knowing that the Leap Motion has a powerful accuracy could allow the development of detailed gestures and, perhaps, this could lead to fewer changes in the WIMP interfaces.

5. Evaluating Leap Motion as a controller for the New Digital Music Instruments.

In [(Silva, et al. 2013)] the authors study the use of the Leap Motion as a way to control a Digital Music Instruments or, in order words, how to play music using the controller.

Digital Music Instruments are, as the name implies, a program that maps certain sounds to a pre-determined input[1]. These software exist in many shapes and forms. One example is a software installed in a computer that requires no specific input tool, normally resorting to the keyboard to work. In these, when the user presses a mapped key, the appropriate musical sound note is reproduced. In other examples, the software requires a specific input device to work[2], being the *modus operandi* similar to the last example. In other cases, the programs come already incorporated inside a machine, modeled similar to the original instrument, simulating the real experience[3].

The experiment was divided in two components: the first was to determine how many real piano-playing gestures the Leap Motion was able to detect and identify and the second was to develop an application that played the piano with the input provided by the device.

In the first task, the authors were able to detect all the movements and gestures performed with the hands to play the piano, namely "palm position, orientation and aperture, fingers length, width and orientation". However, the device was unable to "capture the position of the junction points and segments from the whole arm", which was expected since the Leap Motion premise was only to detect the hand and respective fingers.

In the second task, an interface was developed mimicking a piano. To do this a piano keyboard was drawn in a glass board, large enough to include the basic keys, and placed a certain height above the device. When the user placed is finger/fingers on a part of the glass, in other words, when the user played a certain key, the Leap Motion would read its position, and other body elements and send this input to the application responsible for receiving this input and playing the right sound. This software was also developed at this stage.

In the end, the authors were able to perform the specified experience, not without running into some technical aspects that could be further improved. The Leap Motion controller was lauded as being a good device for controlling Digital Music Instruments, due to its "gestures repertoire". However, several faults were also found, namely in occlusion and loss of tracking. When two fingers were too close the device was unable to distinguish between them and, when suddenly detecting the "lost" finger, the system would play a wrong note.

---

[1] Some examples of Digital Music Instruments: http://www.pcmag.com/slideshow_viewer/0,3253,l=205861&a=205861&po=1,00.asp (accessed 7 February, 2014)

[2] An example: http://www.pcmag.com/slideshow_viewer/0,3253,l=205861&a=205861&po=3,00.asp (accessed 7 February, 2014)

[3] Another example: http://www.pcmag.com/slideshow_viewer/0,3253,l=205861&a=205861&po=7,00.asp (accessed 7 February, 2014)

2.2 Evaluating pointing devices

In this section several patterns of evaluating pointing devices will be presented, including the one that will be used to perform the experiment described in this document. Special attention will also be given to (MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)'s experiment where the authors propose a set of accuracy measures related to the movement performed by the pointer. This would allow this dissertation to have a more detailed characterization of the movement.

2.2.1 Fitts's Law

Fitts's Law, published in 1954, is named after its creator, Paul Fitts, and is a mathematical model of the human psychomotor behavior. According to [(Zhao 2002)], the law is a "formal relationship that models speed/accuracy tradeoffs in rapid, aimed movement". In human-computer interaction, this law can be used to assist in the design of Graphical User Interfaces (GUI) or evaluate the performance of a pointing device, telling how long a user, using a pointing device, will take to reach a target.

This law can be mathematically represented by **(1)** and further complemented by **(2)**.

$$\text{Movement Time} = \mathrm{a} + \mathrm{b} * \mathit{Index\ of\ Difficulty} \tag{1}$$

$$\text{Index of Difficulty} = \log_2\left(\frac{2A}{W}\right) \tag{2}$$

In other words, as seen in **Figure 3**, the time the user takes to move the pointer from a starting point to the center of a specific target, having the target W width and located at A distance, is a logarithmic function of the spatial relative error. The constants "a" and "b" vary depending on the device used.

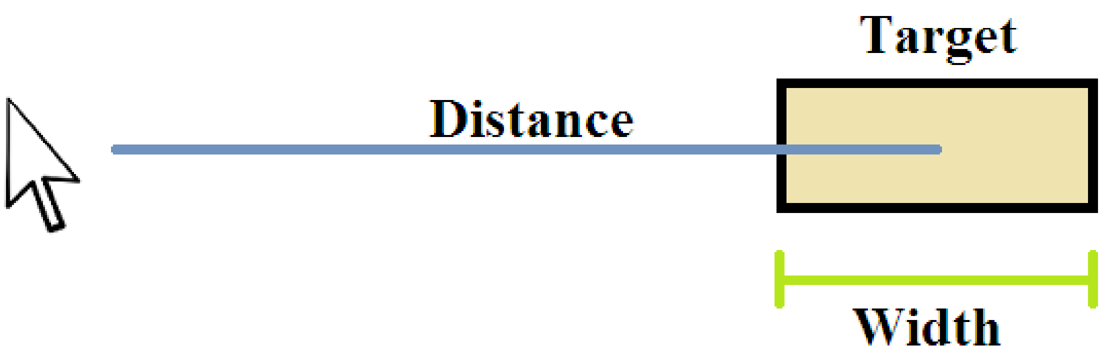

Figure 3: Distance to and Width of the target.

This can be translated as "The farther and small the target is, the harder is to point at it". The contrary, "The closer and bigger the target is, the easier is to point at it", also holds true.

While useful, this law also has its weak points. First, the law makes an (incorrect) assumption that movement is always rapid and aimed, which results in it being represented as a straight line that knows exactly where the center of the target is. This, however, is something that we, humans, are not able to do. We are unable to draw a straight line without support and even less tell the center of something with enough precision.

Second, only the target width is taken into account in the law, never the height, which means that the movement can only be predict in one dimension. Fortunately, this restriction can be easily surpassed, as seen in **Figure 4**, since it should always be (theoretically) possible to draw a straight line between the start point and the target, converting this problem into one dimension.

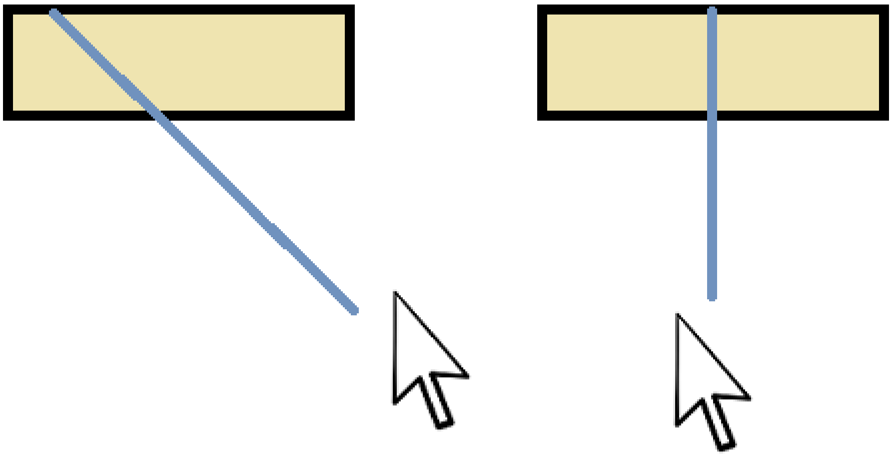

Figure 4: It's always possible to draw a straight line between the start point and the target

Note that, on **Figure 4**, the line doesn't stop at the target's center (on the left example it doesn't even cross it) since it's possible to overshoot, undershoot or completely miss a target center. On the left example, to use Fitts' Law we would have to consider as the center the middle of the line segment formed by the limits of the rectangle that were crossed by the line.

2.2.2 ISO 9241-9:2000

Also known as "Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 9: Requirements for non-keyboard input devices" [(International Organization for Standardization 2000)] is a proposed methodology that, when performed, will evaluate the device performance as a pointing device. Certain subjective elements, as comfort, are also determined during this evaluation.

These tests must be performed by humans as the subjective aspects of the evaluation cannot be calculated; it depends solely on people's opinion. This requires gathering volunteers interested in performing the experiment, either for a fee or for free, with no connection to the project in order to avoid influencing the collected data.

The results attained are both quantitative, resulting from the data collected and formulas used to calculate certain parameters, and qualitative, resulting from the survey that is done to every user who performed the evaluation.

There are several tasks (pointing, selecting, dragging) in which the device can be evaluated. These can be executed several times in order to gather more data. The difficulty of each task can also be changed. As seen in Fitts' Law (which can be find in **section 2.2.1**) changing difficulty means increasing or decreasing the distance between the start point and/or the target width.

In terms of quantitative results, it is possible to determine the **Performance,** also called **Throughput**. When compared to the same value on other devices it can tell which device is the best (has the higher performance).

To calculate **Throughput** (unit: bits per second) one must divide the Index of Difficulty (unit: bit) by the Movement Time (unit: second), according to **(3)**. The ID (Index of difficulty) and MT (Movement Time) are an average of all the trials performed for a single task.

$$\text{Throughput} = \frac{\text{Index of Difficulty}}{Movement\ Time} \tag{3}$$

The Index of Difficulty comes directly from Fitts' law. However, the standard recommends the use of the Effective Difficulty Index (IDe) in order to accommodate the spatial variability observed in responses.

This would turn the ID equation of Fitts' Law into **(4)**.

$$\text{IDe} = \log_2\left(\frac{De}{We} + 1\right) \tag{4}$$

In **(4)**, *De* stands for effective distance, which means the distance traversed by the pointer along the task axis (task axis is a straight line from the actual pointer's position to the center of the target) after the task is over while *We* stands for effective width, which is distance between the pressed point inside the target and its center, being positive when an overshoot occurs and negative when an undershoot occurs. It can be calculated through **(5)**, where *SDx* is the standard deviation in distance (represented as "x") between the pressed point and the center of the target over a block of trials using the same D and W

$$We = 4.1333\ * SDx \tag{5}$$

In terms of qualitative results, a questionnaire is made to every subject participating on the experience in order to determine the subjective parameters, important to define if the device can be used by humans on a common day basis. The questionnaire and possible answers can be found on **Table 1**. The answers are selected from five possible values, as seen in column "Answers" from **Table 1**. For example, on question 2 selecting the middle "X" meant that the smoothness during the operation was moderate while selecting the "X" at its left would mean "rough".

Table 1- Questionnaire and possible answers

Number	Question	Answers
Question 1	Force required for actuation	Very Uncomfortable -X – X- X- Very comfortable
Question 2	Smoothness during operation	Very rough – X – X – X – Very smooth
Question 3	Effort required for operation	Very high – X – X – X – Very low
Question 4	Accuracy	Very inaccurate –X – X – X– Very accurate
Question 5	Operation speed	Unacceptable – X – X – X – Acceptable
Question 6	General comfort	Very Uncomfortable –X – X – X– Very comfortable
Question 7	Overall operation of the input device	Very difficult (to use) –X – X – X– Very easy (to use)
Question 8	Finger fatigue	Very high – X – X – X – Very low
Question 9	Wrist fatigue	Very high – X – X – X – Very low
Question 10	Arm fatigue	Very high – X – X – X – Very low
Question 11	Shoulder fatigue	Very high – X – X – X – Very low
Question 12	Neck fatigue	Very high – X – X – X – Very low

The standard ISO 9241-9:2000 is now deprecated, being substituted by ISO 9241-400:2007 [(International Organization for Standardization 2007)]. However, this dissertation ended up using the first standard was it not possible to acquire the most recent one.

2.2.3 ISO 9241-400:2007

Also known as "Ergonomics of human--system interaction -- Part 400: Principles and requirements for physical input devices", this standard, similar to the last, gives guidelines for physical input devices in interactive systems, like keyboards, computer mice, touch sensitive screens and gesture controlled devices.

While more recent and updated, when compared to the standard to be used on this dissertation, it will not be used. Doing this will not prove a problem since many authors before, [(Natapov, Castellucci and MacKenzie 2009), (McArthur, Castellucci and MacKenzie 2009), (Natapov and MacKenzie 2010), (Cheong, et al. 2011)], have used the deprecated version on their experiments.

2.2.4 MacKenzie, Kauppinen and Silfverberg proposed accuracy measures

In [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)] the authors propose a set of seven new measures claiming that is difficult evaluating different tasks among so many varied systems.

The new measures proposed are **Target Re-entry**, **Task Axis Crossing**, **Movement Direction Change**, **Orthogonal Direction Change**, **Movement Variability**, **Movement Error** and **Movement Offset**. These measures, according to the authors, capture aspects of the movement during a trial, contrary to Movement Time which is based on a single measurement per trial. The first four measures deal with the pointer path discrete events while the last three deal with continuous measures.

1. **Target Re-entry**

Target Re-entry (TRE) measures the number of times the pointer enters the target region (area of the target), leaves it and re-enters again, as seen in **Figure 5**. The blue line is the trajectory performed by the pointer.

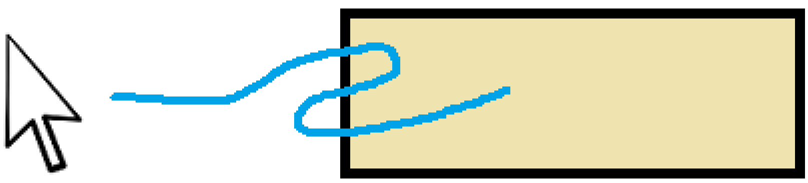

Figure 5: Target Re-entry

The result is presented as percentage, dividing the number of target re-entries occurred by the total number of trials, as seen in **(6)**.

$$\text{TRE} = \frac{Re{-}entries}{number\ of\ trials} \tag{6}$$

2. Task Axis Crossing

Task Axis Crossing (TAC) measures the number of times the pointer crosses the task axis (recalling, task axis is a straight line from the actual pointer's position to the center of the target) when trying to reach the target, as seen on **Figure 6**Figure 3. As an example, the trajectory (red line) crosses the task axis (blue grayish line) only once in this trial.

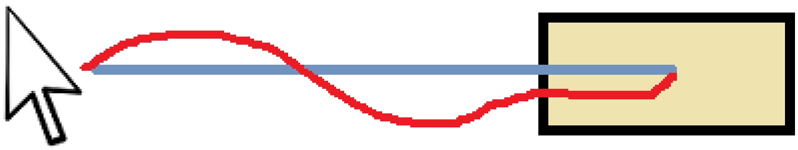

Figure 6: Task Axis Crossing

The result is the number of times the pointer crosses the task axis per trial or per cm of pointer movement.

3. Movement Direction Change

Movement Direction Change (MDC) measures the number of times the pointer’s trajectory changes direction relatively to the task axis. For example, in **Figure 7**, there are two changes in the movement direction.

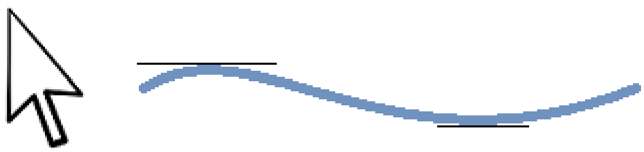

Figure 7: Movement Direction Change

The authors do not give a specific unit for the result, so the dissertation will assume the result is the number of changes of the movement direction per trial.

4. Orthogonal Direction Change

Orthogonal Direction Change (ODC) is very similar to Movement Direction Change. It measures the number of times the pointer's trajectory changes direction along a perpendicular axis to the task axis. For example, in **Figure 8**, there are two changes.

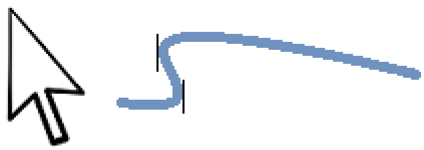

Figure 8: Orthogonal Direction Change

Similar to the last case, the dissertation will assume the result is the number of changes of the movement direction per trial.

5. Movement Variability

Movement Variability (MV) can be interpreted as how far or close the points that form the pointer trajectory during the trail are from an invisible line parallel to the task axis. It can also be interpreted as the difference of amplitude between each point.

For example, in **Figure 9**, in both trajectories, each point's amplitude don't diverge to far from one another. As such, the MV is low.

Figure 9: Trajectories with low Movement Variability

The same doesn't happen in **Figure 10** where the amplitude between each point clearly varies greatly.

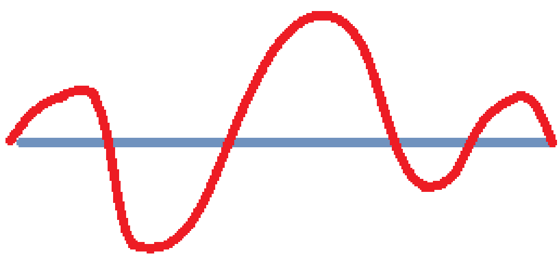

Figure 10: Trajectory with high Movement Variability

To calculate Movement Variability, we assume that the task axis is overlapped with y = 0, as seen in **Figure 11**.

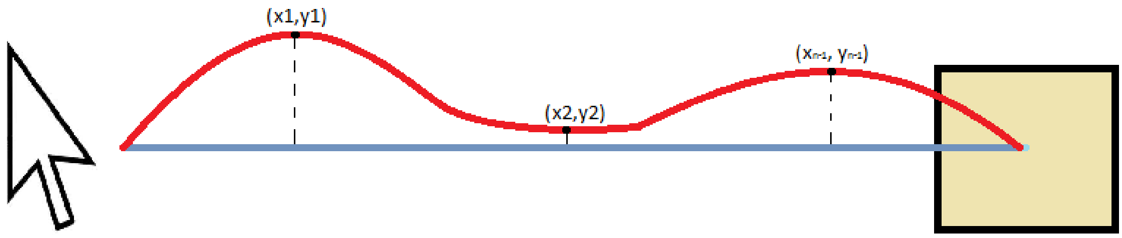

Figure 11: Task Axis overlapping with y = 0

Then, we use **(7)**, where "yi" is the distance between the sample point and the task axis and "ym" is the mean of all sample point distances.

$$\text{MV} = \sqrt{\frac{\sum(yi - ym)}{number\ of\ sample\ points - 1}} \tag{7}$$

A perfectly executed trial (the trajectory overlaps the task axis) should have MV = 0.

6. Movement Error

Movement error (ME) is the deviation (in average) of the points that compose the pointer's trajectory from the task axis. In other words, how far or close the trajectory is from the task axis. It does not matter if these points are above or below it. To calculate, we assume that the task axis is overlapped with y = 0, as seen in **Figure 11**, and use **(8)**, where "yi" stands for the distance between the sample point and the task axis.

$$\text{ME} = \frac{\sum |yi|}{number\ of\ sample\ points} \tag{8}$$

Again, a perfectly executed trial should have ME = 0.

7. Movement Offset

Movement offset (MO) is the mean deviation of the points that compose the pointer's trajectory to the task axis. In other words, this value tells us the direction and distance of the trajectory related to the task axis. The process required to calculate is pretty much the same as the last two. Assume that task axis is overlapped with y=0, as seen in **Figure 11,** and use **(9)** where "yi", once again, stands for the distance between the sample point and the task axis.

$$\mathrm{MO} = \mathrm{mean(yi)} \tag{9}$$

And, of course, a perfectly executed trial should have MO = 0.

2.3 Accuracy evaluation

While not inserted in any of the standards presented before, accuracy is normally evaluated according to the **Error Rate**.

The Error rate is the percentage of trials failed per task and can be calculated using **(10)**. Errors are trials performed by the users where the intended objective is not fulfilled. For example, in a selecting task, an error would occur when a user unintentionally fails to select the intended target. In a dragging task, an error could occur when a user unintentionally drops the target before reaching the intended target.

$$\text{Error Rate} = \frac{number\ of\ failed\ trials}{total\ number\ of\ trials} \tag{10}$$

When evaluating devices, i.e. calculating their **throughput**, only the successful trials are taken into account. However, one can tell that a device with a very low throughput can come from an imprecise device, since users have to take more caution when performing the intended tasks, reducing their speed and increasing the time taken per task, making the error rate a complementary value.

2.4 Summary and Conclusion

In this chapter it was possible to observe that the Leap Motion is, even though quite recent, already applied in many areas, being the most common ones: multimedia, modeling and art. The controller is often used as a complement for the computer mouse, unable to completely replace it. The device is considered the most precise, and cheap, in the market when compared to other gestures based devices. Nevertheless, it's not perfect and several flaws were detected.

An explanation of some evaluation methods was also given, including the one that was used in the context of this dissertation, namely ISO 9241-9:2000. To complement this evaluation MacKenzie, Kauppinen and Silfverberg's proposed accuracy measures were also described.

Nowadays, there are several experiences being performed with this device, some studying its precision, others evaluating if the device is a good controller in a specific field. Yet, there is no evaluations of the controller as a pointing device or as a good mean to interact with a WIMP based interface. It's this gap that this dissertation will try to fill.

Chapter 3

Implementation

This chapter describes the technology used for development, data analysis and data gathering. As it was necessary to develop an application for gathering data in order to mimic the software in Mackenzie, Kauppinen and Silverberg's article, details and specifications will also be included.

3.1 Technologies used

In this section the hardware and software used for developing, data gathering and data analysis will be described and explained. The additional hardware and software used to perform the experiments will be detailed in the next chapter.

3.1.1 Technologies used for development

The technologies used for developing the application responsible for gathering the required data during the performed experiment were two Integrated Development Environment (IDE). For a less morose and a more automatic coding experience, *Eclipse* IDE (version 4.3 or simply "Kepler") was used. This IDE was employed as a source code editor and as a compiling and debugging tool. For developing the Graphical User Interface (GUI), *Processing*, another IDE (and also programming language) was used. The last one was integrated into the first so that development could occur in only one of the tools, simplifying the process. The programming language used was Java.

Java is a programming language and computing platform present in many technologies used nowadays, like computers, cellphones, Internet and many others. One of its main advantages is its compatibility with diverse Operative Systems, achieved through a virtual machine, allowing the same program to be ran almost anywhere without significant (or no) changes.

Eclipse[1] is an IDE developed by the Eclipse foundation and is compatible with both Java, C/C++ and PHP. It also possesses a marketplace, known as "Eclipse Marketplace", where users might download add-ons, allowing its customization. With this, it's even possible to add compatibility to other programming languages that are not available with the regular version. As any IDE, its function is to provide several tools to computer programmers to facilitate software development. These tools can go from compilers, to debuggers or GUI generators.

Processing[2] is also an IDE but with a catch. This open source tool and programming language, according to [(Processing Fondation n.d.)], came about as a way to "... promote software literacy within the visual arts and visual literacy within technology". This way, users could use this IDE to sketch their program using a simple programming language, exclusive to this environment, and without the need to have an extensive background in programming or graphical technologies. This resource has OpenGL integrated in itself and allows the creation of programs with two dimensional (2D), 3D or PDF output. *Processing* is available for several OSes, namely Windows, Linux and Mac OS X.

Processing is built on Java. As such, when merged with *Eclipse*, all of its functionalities were still available and able to be called through code as if working in its IDE.

In the second experiment performed during this dissertation, there was the need to interact directly with the Leap Motion in order to access the device functionalities and information. While it is possible to use other programs who interact directly with Leap Motion just by installing the device, it is not possible to have access to its functionalities without including its SDK or API in the intended project. These makes the Leap Motion a mandatory technology, either in terms of hardware (the device itself) and software (its SDK and/or API). These can be downloaded from the Leap Motion main website[3] and Developers Portal[4].

3.1.2 Technologies used for data gathering

As stated in the last section, one of the tools used to collect (quantitative) data during the experiment was the developed application. With it, it was possible to determine and store how long the user took to perform a trial, if the trial selection was successfully, the path performed by

[1]Eclipse main Website: https://eclipse.org/ (accessed June 16, 2014)
[2]Processing 2 main Website: https://www.processing.org/ (accessed June 16, 2014)
[3]Leap Motion main website, setup page: https://www.leapmotion.com/setup (accessed 9 June, 2014)
[4]Leap Motion Developer's page: https://developer.leapmotion.com/ (accessed June 16, 2014)

the pointer during the trial and other elements. More details on the software capabilities and data gathered will be given in **3.2.1**.

To obtain user's opinions (qualitative data), several surveys were created and presented to each user who participated in the experiments. This survey was developed and performed using several Google Forms and the answers were stored in several Google Spreadsheets. Once the experiments were finished, the results were all assembled into one spreadsheet and then analyzed.

Google Forms is an on-line application that allows users to create forms or surveys using a simple interface. The type of answers the application accepts are varied, going from multiple choices to scale, or from text input to date input answers. Among many other options, the system also as fast dissemination process, creating a link that can be sent to anyone who wish to perform the questionnaire. Lastly, a storage medium (a spreadsheet) is also immediately created so that the query creator may consult the answers given.

Google Spreadsheets, as name implies, is an application, also on-line, where the user can organize/store information in tables. Each cell, line or column of the table can be configured and changed as the user sees fit. These changes can range from altering the size/color/limits of the elements, change the type and size of the font used, increase/decrease decimal places of a number and many others. With the information stored, several calculations can be performed using functions (one common example is calculating the mean of a set of values) or charts might be created.

Google Forms and Spreadsheets are both applications include in Google Docs. Google Docs is a web-based office suite, similar to *Microsoft's* Office but on the web. According to [(Google 2011)], it's possible to create documents, spreadsheets and presentations on-line, allowing it be shared and edited by many users at the same time. Users may also store their files in the cloud through this same service. Nowadays, Google Drive[1], owned by the same corporation, is the cloud storing medium and Google Docs one of its services and functionalities.

3.1.3 Technologies used for data analyses

The technologies required to analyze the data gathered, either qualitatively or quantitatively, during the experiments were a script coded in "R". This script had the function of calculating Target-Re-entry, Task Axis Crossing, Movement Direction Change, Orthogonal Direction Change, Movement Variability, Movement Error and Movement Offset ([(MacKenzie, Kauppinen and Silfverberg 2001)]). Throughput and Error Rate were also calculated. The script is also capable of generating graphics, showing the trajectory of the pointer during each trial.

R, as stated by [(R-Project n.d.)], is "a language and environment for statistical computing and graphics". This language provides statistical techniques, like "linear and nonlinear modelling,

[1]"Meet Google Drive" Webpage: https://www.google.com/intl/en/drive/ (accessed June 16, 2014)

classical statistical tests, time-series analysis, classification, clustering" and many more, as well as the capacity to plot graphics and other elements.

This script[1] was created and provided by Jorge C. S. Cardoso.

3.2 Developed application

In this section details about the developed application for gathering data during the experiment will be given. This includes its specifications, GUI presentation, logical flow and architecture. The created software is already on-line and available to download[2].

3.2.1 Application details

In order to perform the experiment, many elements were taken directly from Mackenzie, Kauppinen and Silverberg's paper. These can be translated into a visual interface that allows the user to perform selecting tasks as if using a WIMP based interface. Behind this visual interface should exist a mechanism able to track the pointer position over time, distance covered by the pointer, width of the targets, distance between the starting point and target center, detection whether a trial (a selection task) was successful or not and the time elapsed for each trial.

Remembering, the objective of a selection task is to select a target as fast and as precisely as possible. This action can be performed by controlling the pointer/cursor onscreen. When above/inside the target, the user must press a certain button (perform a gesture or movement in the Leap Motion) in order to confirm the selection.

The task should be performed as many times as needed, depending on the number of data intended to be collected. The same number must be performed for all the input devices to be evaluated, namely the Leap Motion, touchpad and computer mouse.

One other important element of the application was its sample rate. The sample rate is the number of samples that should be collected in a certain time. In order to match sample rate of the paper, 40 samples per second were used (this means that every 25 milliseconds, one sample had to be read and stored).

---

[1]Data analyses R script:

- http://figshare.com/articles/R_script_Data_analyses_for_Leap_Motion_for_WIMP_based_interfaces_/1061986 (accessed June 21, 2014)

[2]Developed evaluation software:

- https://github.com/jorgecardoso/leapmotionstudy/tree/master/Avaliacao (accessed June 9, 2014)
- http://figshare.com/articles/Leap_Motion_Evaluation_Application/1061984%20 (accessed June 21, 2014)

The created software should also be able to be configurable. This would assure that the application could be used for other kinds of experiments and not exclusively for this one. As such, several parameters can be personalized by the user. These can be found in

Table 2.

Table 2: Application customizable parameters

Parameter	Explanation	Note	Limits
Device ID	The ID of the device currently being evaluated. When ID is 0, the application will communicate directly with the Leap Motion and enable the use of the developed gesture.	The ID is assigned by the person conducting the experiment and has no meaning to the application, except for 0.	The ID must be a value between 1 and 2147483647.
User ID	The ID of the user currently performing the experiment.	The ID is assigned by the person conducting the experiment and has no meaning to the application.	The ID must be a value between 0 and 2147483647.
Right-handed	Boolean indicating if the user is right-handed (TRUE) or not (FALSE).	Used by the application to determine which hand is the dominant one. The dominant hand is the one who will be responsible for the pointer movement.	Like any Boolean, the value must be either TRUE or FALSE.
Number of Circles	The number of targets the experiment has. The targets are circles, whence the name.	This number also determines how many circles will be drawn in the GUI.	This value should vary between 2 and 32.
Circle radius	The radius of every target.	Can be a decimal number.	The radius must vary between 1,0 and 200,0.
Distance between a circle and center	The distance between every target and the	Determines how far the circles are drawn from the application frame	The value must vary between 0,0 and 800,0.

	application frame center.	center. Can be a decimal number.	
Center Offset X	The distance the application frame center should be translated to the LEFT or RIGHT.	To translate to the left values should be < 0. To translate to the right values should be > 0.	Values should vary between -800 and 800. Although bigger or lower values are supported, these could end up being drawn out of the area visible by the user.
Center Offset Y	The distance the application frame center should be moved UP or DOWN.	To move up, values should be < 0. To move down, values should be > 0.	
Random sequence generator	A Boolean. Determines if the order in which targets are selected is random or not.	If not random, the sequence will follow the same pattern as in Mackenzie, Kauppinen and Silverberg's.	Like any Boolean, the value must be either TRUE or FALSE.
Number of sequences[1] per Block	The number of sequences each block will have.	These number will determine how many trials[2] a user must perform[3].	The number could vary between 0 and 2147483647.
Number blocks[4] per experiment	The number of blocks an experiment will have.		

The application also needed to give feedback to the user performing the tasks. This was done either through sound or a visual effect. To let the user know if he/she performed a correct or an incorrect selection a sound was heard. A more violent sound was heard if the selection failed and a more soothing sound if the selection had been executed successfully. Due to the users' feedback, the sound played during the correct selection was removed because it was considered annoying, due to being heard too frequently.

In terms of visual feedback, a message would be displayed in the center of the application any time the experiment started, welcoming the user, when the user was allowed to make a pause, in order to rest and recover from the experiment, or when the user completed an evaluation. The user was allowed to take a pause anytime he/she completed a task. During this time the collection

[1]This term was used in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)]. A sequence is a set of trials.

[2] This term was used in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)]. A trial is a synonym of task.

[3]Total number of tasks = Blocks * Sequences * number of circles.

[4]This term was used in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)]. A Block is a set of sequences.

of data is stopped. To let the user know which circle is the current and intended target a purple "+" sign was drawn upon the specific circle.

In terms of commodity, the user has certain shortcuts, accessible through the Keyboard, available. The "P" key allows the application to enter the Playing Mode, a mode where no results will be saved and created for the user purpose of the users experimenting with the application and gestures without restrictions. The "ESC" key, courtesy of *Processing*, immediately exits the program. Other shortcuts exists but are exclusively for debug.

In terms of dependencies, the application had two needs. The second experiment performed had the user using a specific gesture developed in this dissertation. This required the Leap Motion SDK and Libraries to be installed and reachable by the project. As the software was coded using *Java*, the Leap Motion Java Archive (.jar) was also required. Since, *Processing* was also being used, its Java Archive was also mandatory.

In order to allow portability, the Leap Motion libraries for both Microsoft's Windows and Mac OS were included.

Finally, in terms of Java Runtime Environment (JRE), the application was developed in "JRE 7". The execution environment of the JRE System Library was "JavaSE-1.6" but it is also compatible (requiring no changes) to "JavaSE-1.7".

3.2.2 Graphical User Interface details

In terms of the GUI, the visual interface should have a specific and similar design to [(MacKenzie, Kauppinen and Silfverberg 2001)], as seen in **Figure 12**. This kind of experiment is named by [(International Organization for Standardization 2000)] as a "Multi-directional tapping test" and is used to evaluate "movements in many directions".

Each circle presented in the Graphical User Interface can be likened to a target. For this reason, the number of tasks per trial (or sequence) is equal to the number of circles drawn. As such, the more the number the circles, the more number of trials can be performed in one go.

The purple "+" sign drawn in the GUI represents the actual target the user must successfully select. The order the circles must be selected is indicated by the numbers presented in **Figure 12**, starting in "0" and ending in "15". The circle with number "0" is the trigger of the experiment, only after pressing it the data will start being collected. For this reason the "0" doesn't count as a target. However, this circle is the first starting point.

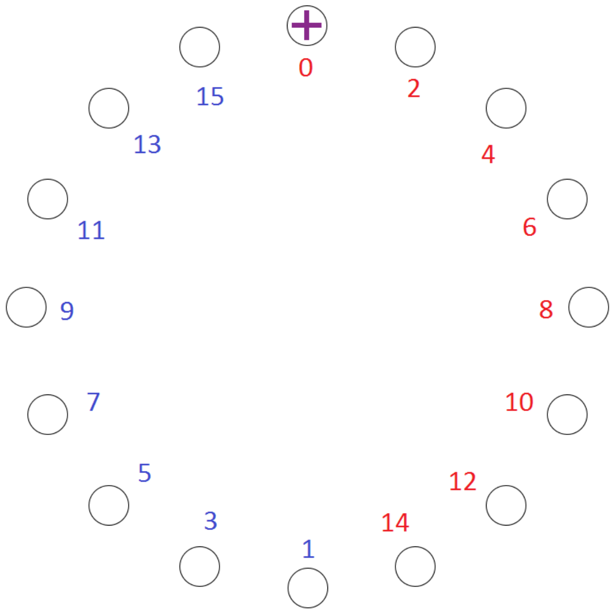

Figure 12: Graphical User Interface design (application screenshot)

The logic behind the order of the circles is as follows:

1. The purple "+" sign always appears at "0". When the user correctly selects this circle the trial starts.
2. The purple "+" sign moves to the opposite circle. This is the next target. The starting point is the point/pixel where the last selection task was completed.
3. When another successful selection is performed the next target is selected. If the opposite circle was already used, select the circle to right of it (clock-wise motion). Once again, this is the new starting point.
4. These actions are repeated until all the circles have acted as a target.
5. When finished, the user may rest. The process restarts again from number **1** until all sequences from all the blocks of the evaluation are completed.

This process is repeated for each evaluation. The number of evaluations is equal to the number of devices to be evaluated in one experiment. In this dissertation, this number is three, namely computer mouse, Leap Motion and touchpad in the first experiment and Computer Mouse and Leap Motion with two different gestures in the second experiment.

The other function of the GUI is to provide visual messages to the user, as seen **Figure 13**. A welcoming message is presented when the software is initiated at the beginning of every evaluation, a message informing that a pause can be taken when every sequence is finished and a "Thank you for participating" message when an a evaluation is completed.

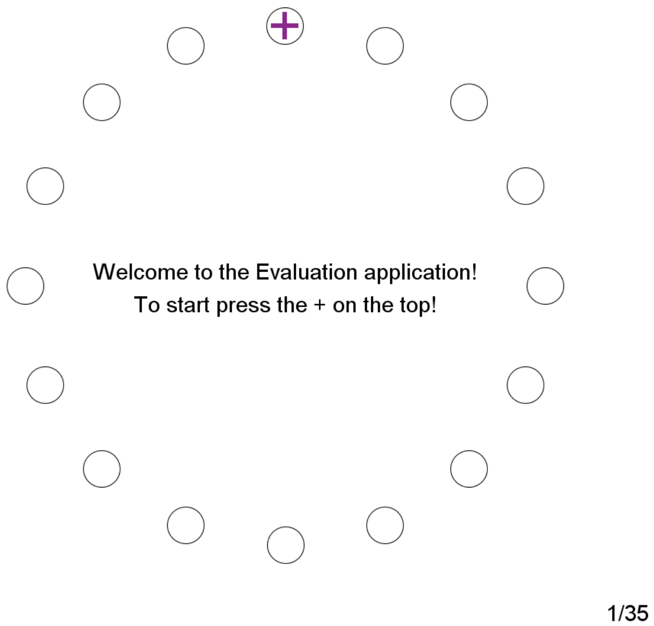

Figure 13: Welcoming message and current sequence number displayed.

The current trial versus ("/") the total number of trials to be performed are also shown in the bottom right corner of the screen.

Finally, the targets have a circular form, contrary to the targets in [(International Organization for Standardization 2000)], which had a square form. This form is due to the Fitt's Law and the Index of Difficulty, more specifically because of the W (width) parameter used when calculating both. If the target is a square the width of the target will vary depending the direction the pointer approached. For example, in a square if the pointer comes directly from above the square, the W will be equal to the height/width of the square. However, if the pointer came from the upper left corner, the W would be √2 of the height/width of the square, as seen in **Figure 14.**

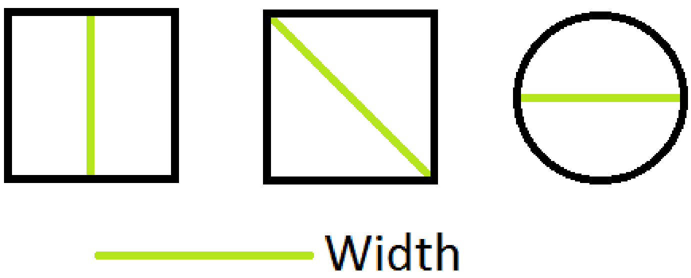

Figure 14: Width might depend on the direction of the pointer came from.

In a circle, no matter the direction of the pointer, the width is always the diameter of the circle. This simplifies calculations as there is no need to calculate or store the width of each target.

3.2.3 Developed gestures

During the development of the application several gestures were created and experimented with. The original idea was to use a movement or gesture developed within this dissertation in a single experiment. However this idea was abandoned and later revived when the second experiment was thought.

During the creation process, several gestures and movements were developed. Some used pre-defined gestures already supplied by the Leap Motion, some were based on existing gestures from other software and others were original. The control of the pointer would be left to the index finger (although other fingers could be used) of the dominant hand. The intention of the gesture was to simulate a click, as if using a mouse, and could be either with the hand controlling the pointer or with the auxiliary hand.

The first options used the Screen Tap[1], Circle[2], Swipe[3] and Key Tap[4] gestures. The Screen Tap gesture was performed by placing one finger over the device, pointing directly at the computer screen, and thrusting it forward, as fast and as precisely as possible, to perform an action (a click in this specific situation), as seen in **Figure 15**. This gesture can be likened to touching or pressing a touchscreen. However, this gesture proved too hard to use and was rejected.

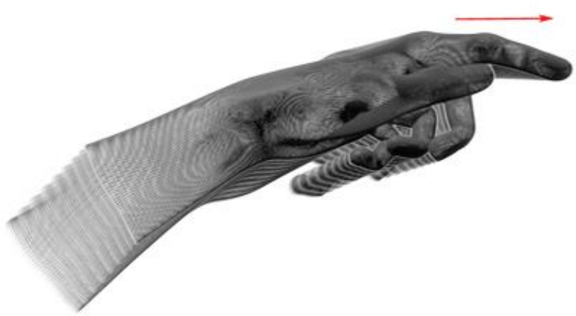

Figure 15: Performing a Screen Tap gesture, Source: [(Leap Motion 2014)]

The Circle gesture could be performed by placing one finger above the device and drawing an imaginary circle, as seen in **Figure 16**. Like the last gesture, this movement was not easily associated with a click and took some time to perform which lead to it also being discarded.

---

[1] Screen Tap API reference: https://developer.leapmotion.com/documentation/csharp/api/Leap.Screen TapGesture.html (accessed June 23, 2014)

[2] Circle API reference: https://developer.leapmotion.com/documentation/csharp/api/Leap.CircleGesture.html (accessed June 23, 2014)

[3] Swipe API reference: https://developer.leapmotion.com/documentation/csharp/api/Leap.SwipeGesture.html (accessed June 23, 2014)

[4] Key Tap API reference: https://developer.leapmotion.com/documentation/csharp/api/Leap.KeyTapGesture.html (accessed June 23, 2014)

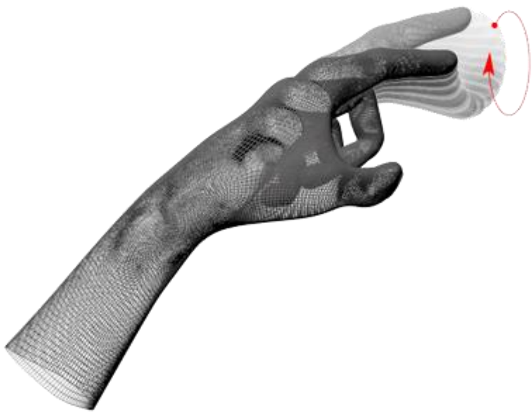

Figure 16: Performing a Circle gesture, Source: [(Leap Motion 2014)]

The Swipe gesture was performed with auxiliary hand, the control of the pointer movement would remain with the dominant hand, by swiping it above the Leap Motion, as seen in **Figure 17**.

Figure 17: Performing a Swipe gesture, Source: [(Leap Motion 2014)]

Unfortunately, this gesture also ended up being discarded due to technical reasons. Since this motion is a continuous gesture, it was analyzed several times by the device without the programmer having a say in the process making it difficult to find a specific moment where to trigger the click. Also, the gesture was detected by finger, which means that depending on the number fingers used by the users, a swipe gesture could end up being detected five times. This would be translated in five clicks, four of them probably wrong. One other problem was that sometimes, when performing the gesture, the auxiliary hand would overshadow the dominant hand, making the device losing track of the index finger, causing the pointer to jump onto a random position in the screen. For all these reasons, the gesture was abandoned.

The last gesture is Key Tap, which is performed by the auxiliary hand, by placing one finger above the device and moving it up and down as if pressing a computer mouse button, as seen in **Figure 18**.

Figure 18: Performing a Key Tao gesture, Source: [(Leap Motion 2014)]

This gesture is also likened to playing a piano key. The user presses it down and, when finished, the fingers returns to its starting position. Contrary to the others, this gesture was considered a good option.

The next two options created both used the device "Touch Zone"[1] functionality and only required one hand to be performed. The touch zone is an invisible plane located above the device and perpendicular to it. The device is able to determine how far the finger is from it and perform an action based on that value. As seen in **Figure 19**, the Touch Zone can be subdivided in three zones: the Touch surface, positioned in the middle of the next two zones, the Hovering zone, the zone between the user and the Touch surface, and the Touching zone, the remaining space after the Touch surface.

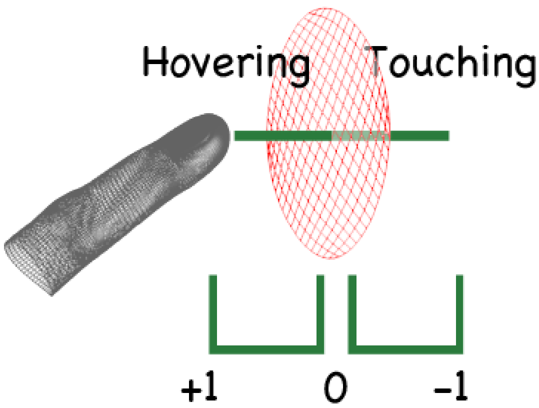

Figure 19: The Touch Zone, Source: [(Leap Motion 2014)]

[1]Touch Zone API reference: https://developer.leapmotion.com/documentation/java/devguide/Leap_Touch_Emulation.html?highlight=touch%20distance (accessed June 23, 2014)

Implementation

In this first option (called "Touch Zone Direct" for convenience), when the index finger controlling the pointer crossed the Touch surface and was "inside" the Touching zone, a click would occur. In the second option (called "Touch Zone Indirect" for convenience), the contrary would happen, when "outside" the Touching zone (inside the Hovering zone) a click would occur. Both these options were considered good and possible controlling methods.

The final option developed is an original gesture and uses the Leap Motion finger detecting capabilities. The objective was to simulate the movement humans make when grabbing something. The user would place his/her auxiliary hand opened above the device and curl the fingers into a fist to perform a click. In other words, the user showed all the fingers when not performing a click and hide all fingers (when curled in a fist the Leap Motion is unable to detect them) when performing a click. Due to the motions required to execute it, it was named "Grabbing Gesture".

This option was also deemed worthy as a controlling method. A measure for stopping the pointer from move when the user had is fingers showing was also implemented. This way, we hoped that when the user was trying to perform a click, less effort would have to be placed in maintaining the position of the dominant hand who was controlling the pointer movement.

Since many options had been created and there was only need for one, one informal experiment was performed. In this, the users would try using all the controls methods (Screen Tap, Key Tap, "Touch Zone Direct", "Touch Zone Indirect" and "Grabbing") several times, with the developed application, to see which they liked the best. As the Screen Tap gesture seemed an intuitive movement, it was given a second chance in this experiment.

The users were then asked to order their favorite control method from one (favorite) to five (most disliked). The results of the survey can be seen in **Table 3**.

Table 3: Informal gesture survey results

Place	Gesture name
First (favorite gesture)	"Grabbing"
Second	Key Tap
Third	"Touch Zone Indirect"
Fourth	"Touch Zone Direct"
Fifth (most disliked gesture)	Screen Tap

Based in this results, the "Grabbing" gesture was chosen as the alternative gesture to be used in the second experiment.

Being this a new gesture, several issues arise when testing this motion. Due to user feedback, this one important change took place. Users declared that keeping their auxiliary hand open during all the experiment was painful. As such, the trigger for performing a click was inversed: the user would place is hand closed above the device and open it, showing all the fingers, when wanting to perform a click. The hand would have to be closed again before proceeding to another task.

3.2.4 Application architecture

The developed application is able to perform its purpose thanks to two (sometimes three) threads and two listeners, as seen in **Figure 20**.

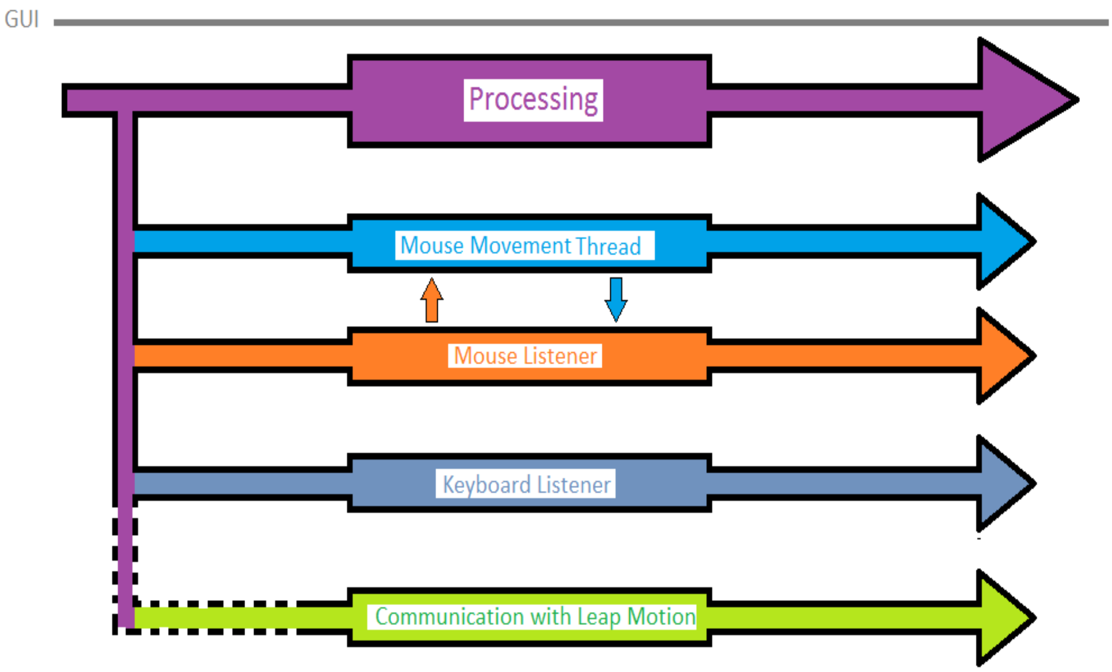

Figure 20: Application Architecture

The main thread is tasked with running *Processing*. Through the *Processing*'s "setup()" function, executed only once, this thread is responsible for setting all the initial parameter the software may require. Other important task is start the thread responsible for storing the pointer movement ("Mouse Movement Thread") and placing the required listeners ("Mouse Listener" and "Keyboard Listener") to detect a click or receive input from the keyboard. Whenever "DeviceID" is set to "0", it is also responsible for starting a new thread that will communicate and parse all data from the Leap Motion.

After this, the main thread is responsible for keeping the GUI drawn and updated until the evaluation ends. This is performed automatically thanks to *Processing*'s, executing its "draw()" function *ad aeternum*.

The "Mouse movement thread" task is to keep track of the pointer position through time, storing its position and if a click has occurred, forty times per second (defined sample rate). This thread receives input from "Mouse Listener" whenever a click happens. Its other functions involve dealing with the logical process of the application, namely: changing to the next target once a successful selection is performed, keep track of the number of unsuccessful tasks, elapsed time and current sequence and block.

At the end of each sequence, the values must be written to a text file, responsibility performed by this thread. Whenever this writing process is taking place, "Mouse Listener" is informed to ignore clicks and, when finished writing, restart listening.

"Mouse Listener" is the thread responsible for checking if a click has occurred. Whenever this happens, it informs "Mouse Movement Thread" of the occurrence. It will also receive orders to alter its behavior and start or stop performing the click detection.

"Keyboard Listener" is responsible for listening any input received from the Keyboard. Depending on the key pressed, an action is to be taken. This thread is majorly used for debugging but also for activating "Play mode" whenever the user performing the evaluation wishes to acquaint with the application.

Finally, "Communication with Leap Motion" thread is not always activated but is responsible for the communication with the Leap Motion. This thread receives the information captured by the device and interprets it. This process is the one that gives the gesture performed by the user any meaning and performs the respective action. This thread is also responsible for masquerading the input from the Leap Motion in a mouse movement (moving the pointer) and a mouse click (performing a click). This way, both listener and thread are capable of detecting this events even though they are not from a computer mouse.

3.3 Summary and Conclusion

In this chapter, details about the developed application were presented. The application requisites were explained, the Graphical User Interface was shown and justified, the gestures developed and chosen were documented and the application architecture was detailed.

With the application and gesture complete, it's now time to put it to good use. The next chapter will detailed the experiments that were performed to evaluate the Leap Motion, and the other devices, using the developed software.

Chapter 4

Experiment Setup and Flow

In this chapter a detailed explanation of the performed experiment will be given. During this dissertation two experiments were performed. The first was the evaluation of the Leap Motion as a pointing device. The computer mouse and touchpad were also evaluated to compare their results with the Leap Motion.

Based on the results of the first experiment, the Leap Motion was disliked by the user who performed the experiment. In order to determine if this is due to the Leap Motion or to the controlling mode chosen when using the device, a second experiment was performed.

4.1 Apparatus for the Experiment

4.1.1 Evaluating the Leap Motion, touchpad and computer mouse

To perform the first experiment, the developed software was installed in a computer. The machine was an *Apple Mac Mini*[1] with the "Mac OS X" (version 10.8.3) operative system. The computer was connected a LCD screen, HP L1706 LCD Display[2], through a VGA cable. An adapter[3] compatible with both the cable and computer had to be used to perform the connection between both. The screen resolution was 1280 x 1024 (Width * Height). The resolution is one of the most important definitions of the experiment since it influences the drawing of the GUI and may change the Index of Difficulty.

[1]Computer specifications: http://www.apple.com/pt/mac-mini/ (accessed June 9, 2014)
[2]Display specifications can be found here (accessed June 9, 2014)
[3]The specifications of the adapter used can be found here (accessed June 22, 2014)

Experiment Setup and Flow

A keyboard was also present during the experiment. As the Keyboard is only a tool for customizing the software and not to be used to perform the evaluation, its specifications are not worth mentioning (any keyboard could have been used).

Since the experiment also included the evaluation of a computer mouse and a touchpad, one of each had to be used. For the computer mouse a *Genius Xscroll*[1] wired mouse was used. For the touchpad, an *Apple Magic Trackpad*[2] was used. The necessary drivers were already installed in the computer. In terms of definition, the speed of the pointer in the Computer Mouse has set to the Slowest + 2 (the slider should be located above the third line) and in the touchpad it was set to the Slowest + 3 (the slider should be located above the fourth line), as seen in **Figure 21**.

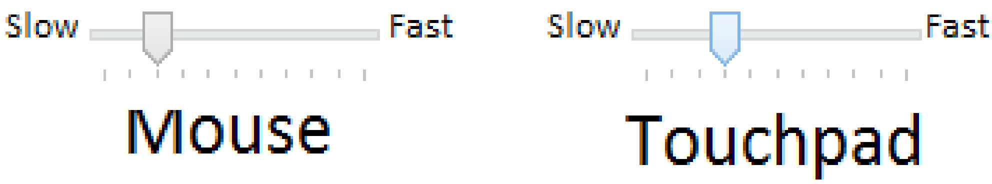

Figure 21: Mouse and Touchpad pointer speed.

One of the most important components is the Leap Motion device. The device was connected to a computer through the USB cable (the long one) that came with it. The necessary libraries for the device to be detected and work had to be download from the Leap Motion main website[3]. In terms of definitions, the tracking was set to "Balanced".

To control the pointer using the Leap Motion, a free commercial software was used, named Touchless[4]. The Touchless is an application developed exclusively for the Leap Motion capable of transforming the fingers and the number of fingers in input. The user can use one finger to control the pointer and the distance of that finger to an invisible plane, perpendicular to the device, to make a click. Using more than two fingers at the same time, will allow the user to activate the scroll. According to the application tutorial, the software has two modes, Basic and Advanced, being the difference between both the number of actions a finger can perform. For example in basic mode, one finger can only move the pointer and perform a click. In advanced mode, a scroll, drag and draw actions are also possible using one finger.

---

[1]Mouse specifications: http://www.geniusnet.com/Genius/wSite/ct?xItem=16529&ctNode=104&mp=1 (accessed 9 June, 2014)
[2]Touchpad specifications: https://www.apple.com/magictrackpad/ (accessed 9 June, 2014)
[3]Leap Motion main website, setup page: https://www.leapmotion.com/setup (accessed June 9, 2014)
[4]Touchless Airspace page for:

- Windows: https://airspace.leapmotion.com/apps/touchless-for-mac/osx (accessed June 21, 2014)
- Max OS X: https://airspace.leapmotion.com/apps/touchless-for-windows/windows (accessed June 21, 2014)

In order to install Touchless, an Airspace (Leap Motion Store) account and application are needed. The download and installation are performed automatically by this software. To use Touchless, Airspace must always be started first.

Finally, a connection to the internet was required as the surveys to perform during the experiment were stored online.

4.1.2 Evaluating the Leap Motion using different gestures

The apparatus for the second experiment was the same as the first, with an exception. This time, the touchpad was removed from the experiment. Only the computer mouse (due to being the favorite device of the first experiment) and Leap Motion were used.

To control the pointer onscreen the Leap Motion used two different methods. The first remained the same, being Touchless the responsible for this control. In the second, the gesture developed in **3.2.3** was charged of doing the control. To activate this gesture, the software definitions, namely the “DeviceID” parameter, would have to been set to "0".

4.2 Application customization

Every experiment performed in this dissertation had application parameters set to the ones present in **Table 4**.

Table 4: Default parameters for every evaluation

Parameter	1st Experiment	2nd Experiment
Number of Circles	16	
Circle radius	24.5	
Distance between a circle and center	340.5	
Center Offset X	0	0 / 200*
Center Offset Y	0	
Random sequence generator	False	
Number of sequences[1] per Block	5	
Number blocks[2] per experiment	8	7

[1]This term was used in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)]. A sequence is a set of trails. A trial is the number of selection tasks a user must perform and equals the number of targets less one.

[2]This term was used in [(MacKenzie, Kauppinen and Silfverberg, Accuracy Measures for Evaluating Computer Pointing Devices 2001)]. A Block is a set of sequences.

The parameters that were not present in **Table 4** vary too frequently. "Device ID" must be changed to the respective value every time an evaluation is performed, totaling 3 times per user. "User ID" must be unique and changed every time a new user performs the experiment. The same happens to "Right-handed" that must be set accordingly to the user.

There are some exceptions, namely "Center Offset X" and "Number of blocks per experiment", that should be referenced.

"Center Offset X" is used in order to translate the GUI a little to the right in order to simplify the process of using the developed gesture. As the gesture uses both hands, the translation ensures (or at least should) that there is enough space to avoid the overlapping of the left and right hand. However (*), this value should only be set when the evaluation of the Leap Motion is being performed using the developed gesture. When evaluating the computer mouse, touchpad and the Leap Motion (through Touchless) the "Center Offset X" should be 0.

"Number of blocks per experiment" also varies between experiments. This is due to the number of times the Leap Motion is used in each experiment. As the device demands some physical effort and takes more time to use than the other devices, a reduction had to be done from the first experiment, where the device is only used once, to the second, where the device is used twice. Otherwise the experiment would take more time and effort than the users would be willing to spend.

While constant during the experiments, the parameters "Circle radius" and “Distance between a circle and center" have a little particularity that should be referred. The values assigned to these only have a meaning when the screen size and resolution (1280 x 1024) are the same as the ones specified Otherwise, the drawn distance and size of the circles might end up being different, changing the Index of Difficulty and Fit's Law. If these are different, then the circles radius must be equal to 13 millimeters and the distance between the circle center and the center of the applications frame must be 180mm. The Index of difficulty between the center of one circle (stating point) and the center of opposing center (target) must be equal to 3.8[1] bits.

4.3 Procedure

4.3.1 Evaluating the Leap Motion, touchpad and computer mouse

Before the experiment started, all the apparatus must be mounted and ready to be used. The software was ready to be initiated at any time and the surveys were loaded using a browser. A

[1]According to [(International Organization for Standardization 2000)] a selection task with a Index of Difficulty lower than 4 is considered a low precision task.

unique "UserID" was inserted into the created an application. The settings of the mouse, touchpad, Leap Motion and Touchless were set accordingly.

When the user arrived, the objective of the experience was explained to him. The devices to be evaluated were presented to the user and explained whenever required. The Leap Motion commonly required a more detailed explanation since only a low number of the users were aware of it and its functionalities. Whenever this happened, the Visualizer, an application included in configuration panel of the device that showed the user's hands and fingers, was used to let the user get acquainted with the process. The most common errors (fingers or hands to close to each other, overlapping hands) were also explained.

Before starting the first evaluation, the user was asked to fill a survey[1] in order to determine his/her experience with a computer and the devices being evaluated. Gender and Age were also questioned. While not contained in the survey, the user was asked if he/she any diseases / limitations in terms of hands, fist, arm, shoulders and back that could affect the experiment. All the information collected was for statistical analysis and no personal or individual information would be revealed.

Every user who participated in the experiment had to evaluate the devices in a different order, as to assure that the order of the devices had no influence in the results. Since three devices were to be evaluated, six different possible ways exists to perform the experiment, as seen in **Table 5**.

Table 5: Different device evaluation order for the first experiment

Possible different sequences of device evaluation		
Leap Motion	Computer Mouse	Touchpad
Computer Mouse	Touchpad	Leap Motion
Touchpad	Leap Motion	Computer Mouse
Touchpad	Computer Mouse	Leap Motion
Leap Motion	Touchpad	Computer Mouse
Computer Mouse	Leap Motion	Touchpad

Each user performed one of the available sequences, starting from the first line of **Table 5**. When all the sequences were exhausted, the order could be repeated starting from the first line once again.

[1]"Demographic and Computer Literacy" survey can be found in **Demographic and Computer Literacy survey**.Questionnaires

Experiment Setup and Flow

When the order of devices has chosen, the first device would be cleaned and connected to the computer, all the others were disconnected, except for the computer mouse and Keyboard, who would be used to start the application and change the respective configurations.

Before initiating the software, the user was asked which hand was is dominant hand. Alternatively, it could also be asked which hand the user preferred to control the mouse, as some users employ a different hand than the dominant for controlling the mouse. The selected dominant hand would be the hand whose index finger controlled the pointer when using the Leap Motion. The "DeviceID" would also be change to match the one being evaluated.

After these steps, the application would be started and explained to the user, including what is a selection task, objectives, the number of sequences (or trials) to perform and that a sound would be played whenever the user performed a unsuccessful selection or a successful one (this was later removed due to feedback by the users that the sound played when a selection was successful was annoying and was played too many times).

"Playing mode" was then activated ("P" key was pressed) and three selection tasks were performed so that the user could better grasp his/her intended task. Next, the user himself/herself would be given some time to get acquainted with the application and the task at hand. If the Leap Motion was being used the steps required to move the pointer and perform a click would also be explained. Possible errors that might show during the experiment (Touchless activating scroll mode, device suddenly stop tracking) were explained and possible solutions were given.

Once the user felt prepared, the real evaluation could start. The application was restarted and the computer mouse disconnected (unless it was the one being evaluated). The user was told that the evaluation was evaluating the device and not to himself/herself and was asked to perform the evaluation as fast and precise as he/she could while avoiding performing errors. The user was also told that after each sequence he/she could make a pause. This pause was not mandatory and if the user felt like it, he/she could ignored it and proceed with the evaluation. The application would present a message any time this pause could be taken.

The user was also asked if his/her posture during the evaluation could be registered, in order to try to understand a pattern or a correct position for using the devices, and to give any feedback he/she felt was important related to the evaluation or any pain/discomfort felt.

While the user was performing the experiment, his/her comments and posture were annotated, any exception (using the device in an unexpected form, for example) performed was also registered.

When the evaluation of the current device is finished, the user was asked for any pain/discomfort he/she felt and asked to fill a survey[1] relating to the experience of using the device.

The process for the other devices is similar to the one described before. When all the devices were evaluated, the user was asked to answer one last survey[2] in order to determine which device was his/her favorite. By favorite, it meant which device the user preferred using every day for interacting with WIMP interfaces. Once finished, the user was thanked for his/her presence and help.

4.3.2 Evaluating the Leap Motion with different control modes

The second experiment performed was very similar to the first and, as such, the same procedure can be followed. However, some changes were required.

Based on the results of the first experiment, the computer mouse was the favorite device and the Leap Motion the least favorite. As such the Touchpad was removed and to discover if the dislike for the Leap Motion has due to control method or the device in itself, the developed gesture was added. As such, the possible order in which the experiment could be performed as changed, as seen in **Table 6**.

Table 6: Different device evaluation order for the second experiment

Possible different sequences of device evaluation		
Leap Motion (Touchless)	Computer Mouse	Leap Motion (Grabbing Gesture)
Computer Mouse	Leap Motion (Grabbing Gesture)	Leap Motion (Touchless)
Leap Motion (Grabbing Gesture)	Leap Motion (Touchless)	Computer Mouse
Leap Motion (Grabbing Gesture)	Computer Mouse	Leap Motion (Touchless)
Leap Motion (Touchless)	Leap Motion (Grabbing Gesture)	Computer Mouse
Computer Mouse	Leap Motion (Touchless)	Leap Motion (Grabbing Gesture)

[1]The questions found in this survey are the same as the ones found in **Table 1**. The same questions were performed for each device. An example of this survey can be found in **Individual device evaluation survey**.
[2]"Device preference" survey can be found in **Favorite device survey (1st experiment)**.

Note, however, that the number of users performing the second experiment was inferior to twelve, not allowing every order to be used at least twice.

In terms of surveys, the number performed remained the same but the survey related to the touchpad was removed. Also, two surveys[1] were performed to the Leap Motion, one for each control mode. The questions made in both remained the same.

For the final survey, where the device preference was asked, several new questions were added[2] in order to find which gesture/control mode the users preferred when using the Leap Motion.

Finally, anytime the developed gesture ("Grabbing Gesture") was evaluated, "Center Offset X" parameter was changed from 0 to 200.

4.4 Participants

There were no restrictions in terms of age, gender, dominant hand or experience with the devices when recruiting participants. It was also tried to find participants that had experience using the Leap Motion, whoever, since the device is still recent, it was impossible to find participants who had the same experience with it as they did the computer mouse or touchpad.

4.4.1 Evaluating the Leap Motion, touchpad and computer mouse

For the first experiment, twelve non-paid users (9 were male, 3 were female) were recruited at *Centro de Investigação Académico da Escola das Artes* although outside personnel and other persons were also allowed. Their ages ranged from twenty-one to sixty, as seen in **Table 7**. No user aged between forty-six to fifty-five years participated in the experiment.

Table 7: Ages of the participants of the first experiment

[21-25]	[26-30]	[31-35]	[36-40]	[41-45]	[56-60]
1	3	5	1	1	1

In terms of computer literacy, every user claimed using a computer every day. In terms of experience using the devices in evaluation, the frequency with each user utilized each device was somewhat more disperse, as seen in **Table 8**.

[1]These surveys are the same as the last section and contain the same questions as **Table 1**. See **Individual device evaluation survey**.
[2]"Device and gesture preference" survey can be found in **Favorite device and gesture survey (2nd experiment) .**

Table 8: Number of users by frequency of use of each device, in the first experiment

	First try	Seldom	Often	Every day
Computer Mouse	0	1	1	10
Touchpad	1[1]	2	3	6
Leap Motion	7	5[2]	0	0

4.4.2 Evaluating the Leap Motion with different control modes

For the second experiment, 9 non-paid users (4 were male, 5 were female) were recruited at *Faculdade de Engenhaira da Universidade do Porto*. The user ages ranged from zero to thirty-five, as seen in **Table 9**. No users between fifteen and twenty years old participated on the experience.

Table 9: Ages of the participants of the second experiment

[0-14]	[21-25]	[26-30]	[31-35]
1	5	2	1

In terms of computer literacy, all users except one claimed using a computer every day. The remaining user said that it used the computer often. In terms of experience of the computer mouse, eight users claimed using a mouse every day, while the remaining one used this device only often. In respect to the Leap Motion, the answers were the opposite. Every user who participated in this experiment had none experience with the device.

4.5 Summary and Conclusion

In this chapter, details about the experimental setup and flow were presented. All the apparatus used to perform both experiments were discriminated and the flow of each was explained in high detail.

Finally, the number of participants and their experience with computer and the evaluated devices were also presented.

With the results gathered from the evaluation described in this chapter, it is now time to analyzed then and extract conclusions. The results of this task will be presented in the following chapter.

[1]While not the first time using it, this user disliked the device and, as such, had little experience with it.
[2]This users claimed already having used the device more than once. However, the frequency with which they used it was low.

Chapter 5

Experimental Results

. In this chapter the results collected by the application or through the performed surveys were analyzed. All the quantitative results find in this chapter were calculated using the methods described in [(MacKenzie, Kauppinen and Silfverberg 2001)], ISO 9241-9:2000 and Error rate. All the formulas and process of calculation were already presented in **Related Work**, namely in section **2.2** and **2.3**.

For the qualitative results originated from the surveys, these were parsed and the mean and standard distribution were calculated, using a confidence interval of 95%. The postures taken and commentaries given by the users, registered during the experiments, are also be described and analyzed in this chapter.

Finally, a set of guidelines, resuming the conclusion taken of this chapter will also be given.

5.1 Quantitative analyses

In order to try to place the devices on equal footing, the first three sequences (the first fifteen trials) were not taken into account when performing this analyses. This is due to the learning curve that the Leap Motion exhibited in the first sequences, as seen in **Figure 22**. In the first three blocks (Block 1, 2 and 3) Movement Time is higher than in the remaining blocks and, as such, we assumed that this is the learning period and will not evaluate it.

Note that is not correct to assume that the learning phase has ended. From block 5 to 7, there is another decrease, indicating a possible continuation of this period. However, in the second experiment, for the same device and control mode (Touchless), the decrease in block 5 to 7 does not happen, as seen in **Figure 23**. For this reason only blocks 4 to 8 have been considered.

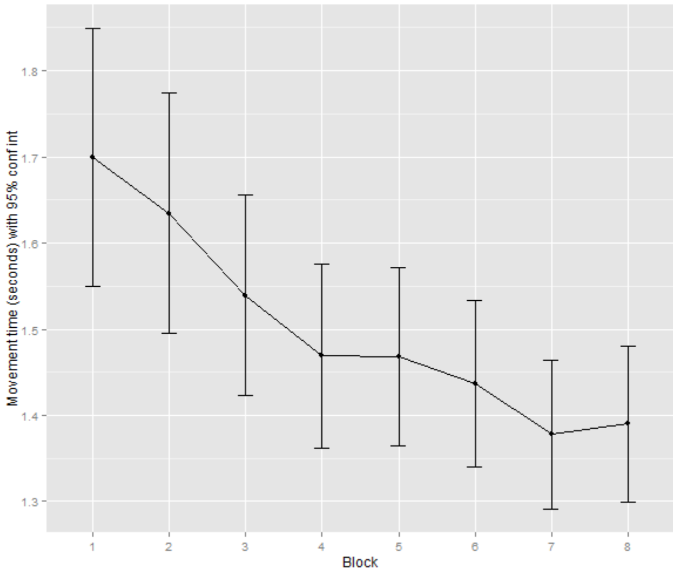

Figure 22: Leap Motion (Touchless) movement time during the first experiment

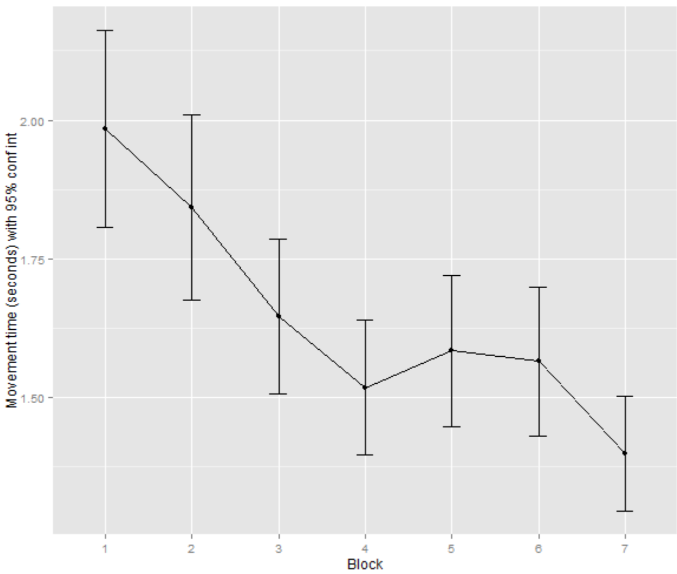

Figure 23: Leap Motion (Touchless) movement time during the second experiment

5.1.1 Evaluating the Leap Motion, touchpad and computer mouse

In the first experiment, the Leap Motion (using Touchless), touchpad and computer mouse were evaluated. The specific values of this experiment can be consulted at **Evaluation results,** section **B1** and **B2**.

In terms of Movement Time, the Leap Motion was one of the devices that required the most time to perform the forty trials. As seen in **Figure 24** (ignoring blocks 1 to 3), Movement Time for the Leap Motion is higher than twice the computer mouse. In other words, the Leap Motion took more than the double the time the computer mouse took to complete its evaluation. Equally drastic is the difference with the Touchpad. These values can perhaps be justified due to the number of times or frequency that users used them before. Being a new device, there's no user yet familiarized to it as, for example, to the computer mouse or touchpad, requiring more time to perform the same tasks.

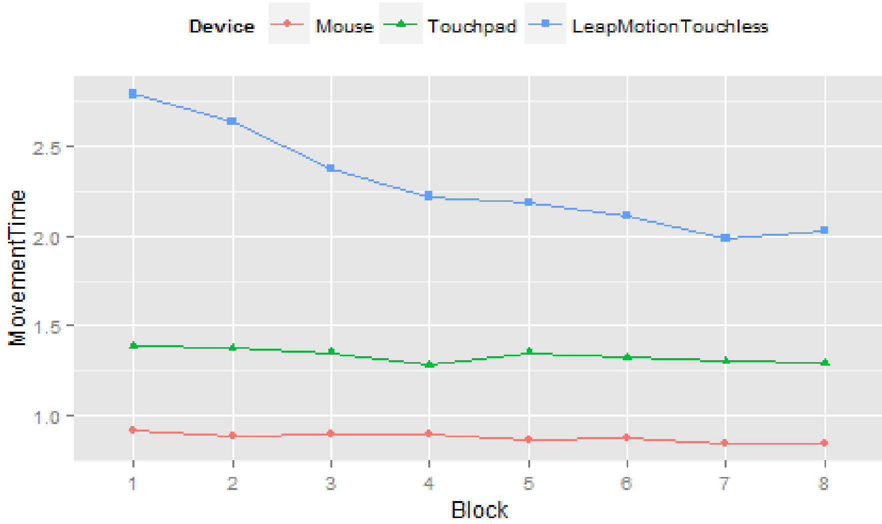

Figure 24: Comparison of Movement Time between the tree devices (1st experiment)

With these results, it not surprise that the Leap Motion is also the device with the worse performance, as seen in **Figure 25**.

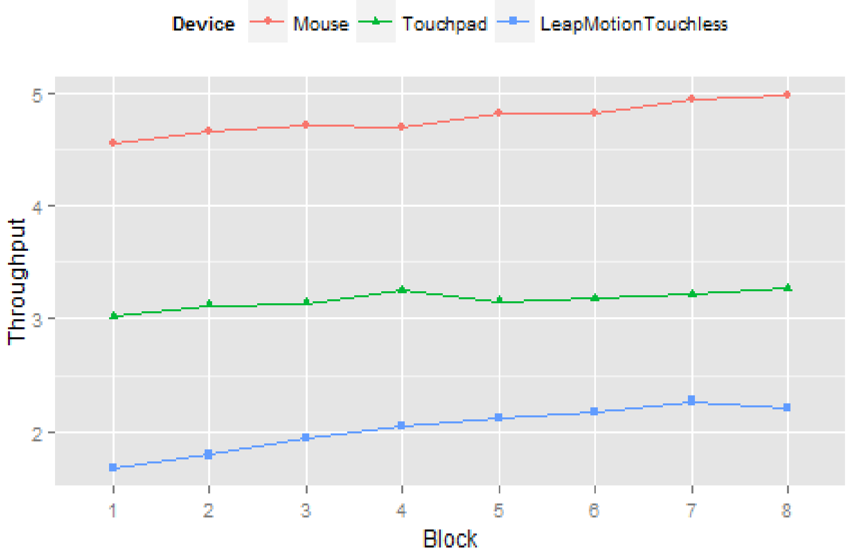

Figure 25: Comparison of Throughput between the three devices (1st experiment)

The Throughput is indirectly proportional to the Movement Time, meaning that the longer a device takes to perform a tasks with the same difficulty the lower its performance will be.

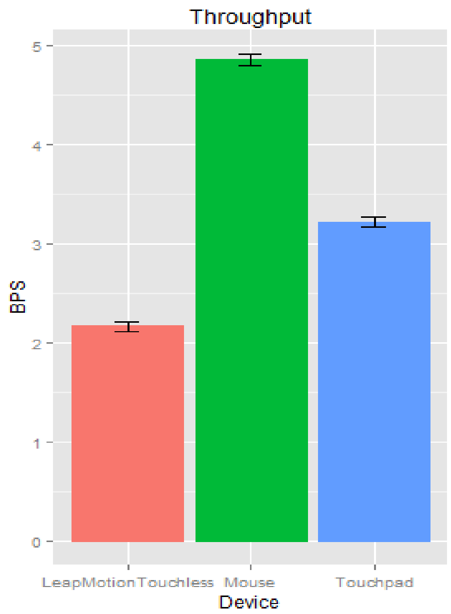

Figure 26: Bar plot of each device Throughput (1st experiment)

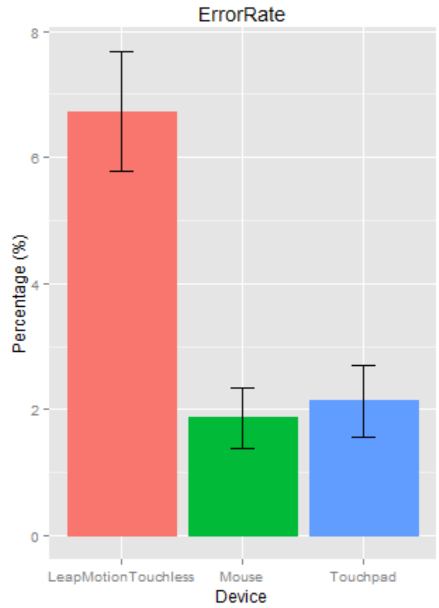

Figure 27: Bar plot of each device Error Rate (1st experiment)

In conclusion, and based in the bar plot in **Figure 26**, with a confidence interval of 95%, the device with the best performance was the computer mouse, followed by the Touchpad and finally the Leap Motion.

In terms of Error Rate, the Leap Motion attained the biggest rate of the three devices, as seen in **Figure 27**. While not high, only six percent, it is three times more than any of the remaining devices. A possible explanation to this problem could be tracking issues. When using the device, several users left their thumb unhidden and detectable by the Leap Motion. To perform a click a user had to penetrate the Touching surface, reaching the Touching zone. When exaggerated, the user not only placed is index finger as well it thumb, resulting in two clicks. Other reason could be the clicking process. When getting close to the touch zone, the user had to keep is finger horizontal, pointing straight at the computer screen, otherwise, when approaching the touch surface, the pointer would translate a little, due to not being straight, ending outside of the target and resulting in a wrong trial.

The computer mouse and touchpad had a very similar error rate but due to the overlap of the confidence interval it is not correct to declare that the computer mouse had an error rate lower than the touchpad.

When analyzing the seven accuracy measures, the results presented in **Figure 28** were reached. By looking at bar plots several conclusions can be extracted.

In terms of TRE, the Leap Motion was the device who had the most difficulty keeping the pointer in the target after reaching it. This could be, once again, the reason the Error Rate is so high. When approaching the Touch Zone, if the finger was not horizontal, the pointer would translate a bit, ending outside the target.

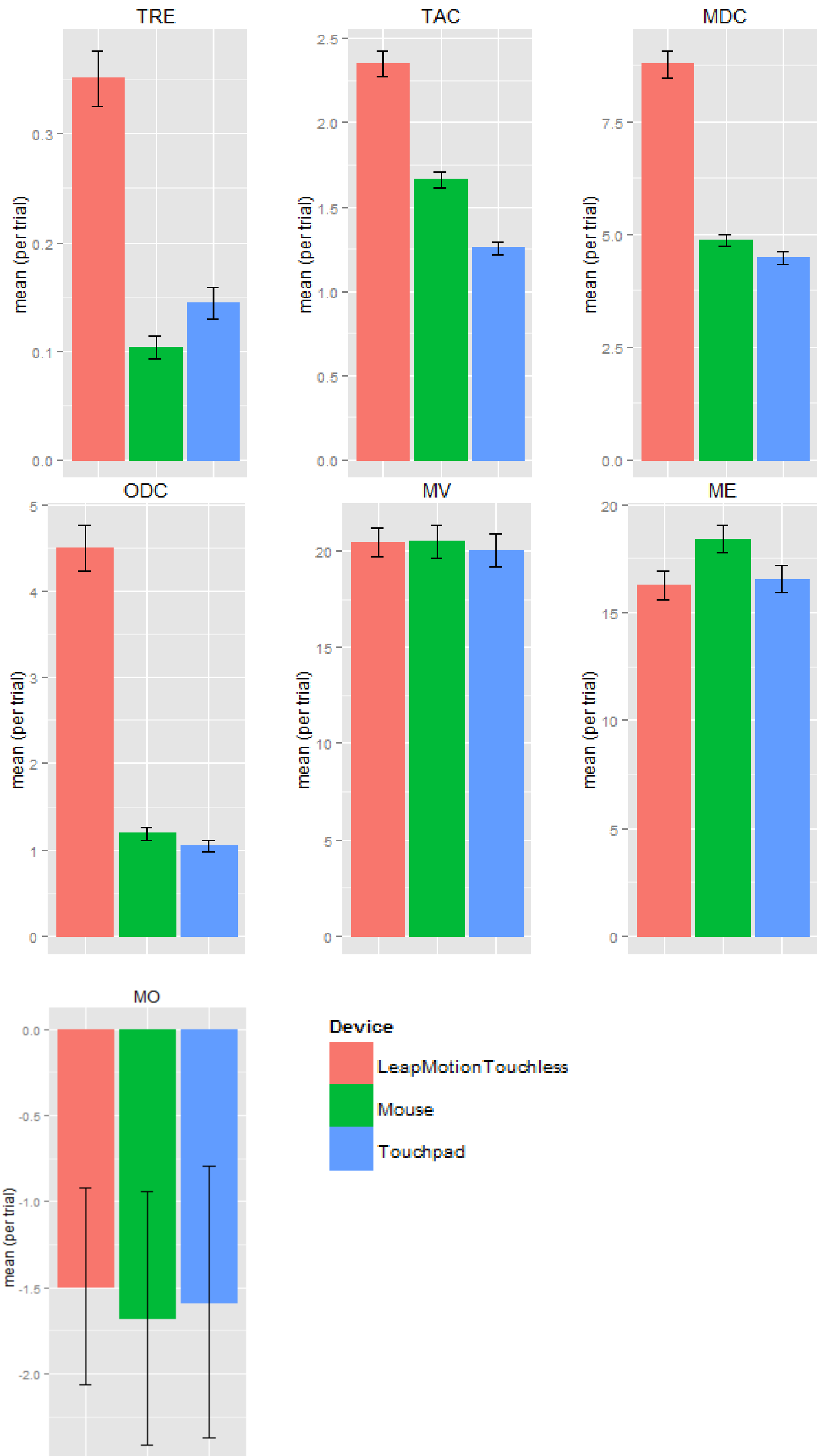

Figure 28: Seven accuracy measures for each device (1st experiment)

Experimental Results

As for the computer mouse and the touchpad, the results presented can be justified due to the overshooting of the target. This a method done naturally by the users when using this device in their computers. When trying to reach an icon far away, the user shoots the pointer in its direction, many times passing through the target, only to slow down, correct the route and perform a successful click.

Analyzing the TAC, once again the Leap Motion presents the highest value. This value could be justified by the natural tremor of the human hand, making the hand shake enough cause it to cross the task axis. As the mean is not that high, one other reason could lie in the task axis' extremes, the starting point and the target center. When starting the task, the user could naturally cross this axis. When reaching the target and trying to point to an exact position, the effort and sudden adjustments could lead to crossing the axis.

After the Leap Motion, the computer mouse is the device with second highest TAC and lastly the Touchpad. The same justification from the Leap Motion could apply to these results. When in the extremes of the task axis, the crossing of it will most likely occur. The reason that the values for this two devices is lower might be due to the overshooting of the pointer. This way the pointer remains for from the axis, only coming close to it in the beginning or end. The difference between the computer mouse and the touchpad might be related to the way these are controlled.

Focusing on the MDC[1], the Leap Motion is the device with highest value, followed by the computer mouse and lastly the touchpad. The human tremor might have influenced the Leap Motion results were shown. Other justification might be related to the TRE and TAC. If the pointer leaves the target area when trying to press it, the direction must be changed to re-enter it. Also, if the task axis is being crossed several times, it means that the direction is also changing frequently in order for this to happen.

Relating to the MDC is ODC[2]. The results between both are similar and the same reasons might apply as well. One interesting thing to note is that to perform the trials several more up/down changes in the direction are required than left/right. Between the computer mouse and the touchpad, the last has the lower ODC (but only by a very small interval of 0.00565).

Focusing more on the movement of the pointer, MV had a very similar value to all the devices evaluated, this means that the trajectory performed by the pointer when controlled by each device it is not that different. In **Figure 29**¸ the mean of MV between each block for each device is not higher 2,5. The reason this might happen is because of the task at hand. To perform a selection task as fast as possible one way is to perform a straight movement (a line) from a start until the target. While the user shakes his/her hand and moves the pointer up/down/left/right,

[1]Remembering, the Movement Direction Change (MDC) deals with "up-to-down" and "down-to-up" changes in the trajectory direction.

[2]Remembering, the Orthogonal Direction Change (ODC) deals with "left-to-right" or "right-to-left" changes in the trajectory direction.

he/she tries to keep a fix trajectory to the target, not allowing for a big range of distance between the two farthest opposing points in the trajectory.

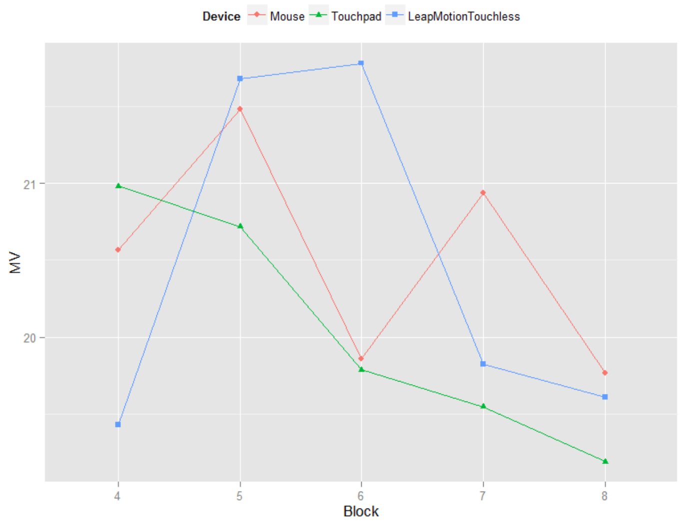

Figure 29: Differences in MV mean for each device per block (1st experiment)

In terms of ME, the computer mouse had the highest value. Due to the confidence interval overlapping, it is not right to assume that the touchpad had a higher value than the Leap Motion. Based on this, we can assume that the trajectory performed by the users when using the computer mouse is far from the task axis by 19 – 22 pixels. This might be due to the overshooting of the pointer that happens in the computer mouse, which is somewhat lower in the touchpad due to the way device is used. This value might be lower in the Leap Motion because the user has less proficiency with it, using less bold moves, trying to keep a straight trajectory to the target. By trying to do this straight line, the distance he/she keeps from the task axis is lower.

Finally, analyzing MO, all the same devices tend to a very similar value, being impossible to tell which one has the highest one. One interesting point to note is that all the trajectory performed using any of the three devices tends to go below (or right) of the task axis.

5.1.2 Evaluating the Leap Motion with different control modes

In the second experiment, the Leap Motion and the computer mouse were evaluated. The Leap Motion was evaluated using two different control methods, the *Touchless* and the developed gesture. The computer mouse was once again evaluated, not with the intention of collecting new data but to allow the new participants to try the Leap Motion and the last experiment favorite device. Once again, for a more details on the experiment values, several tables were added to this document at **Evaluation results**, specifically section **B3** and **B4**.

With the addition of the new gesture it was possible to reduce Movement Time required to perform the tasks at hand, as seen in **Figure 30**. Due to the reason presented at the start of this chapter, the analyses will ignore the first three blocks of the experiment.

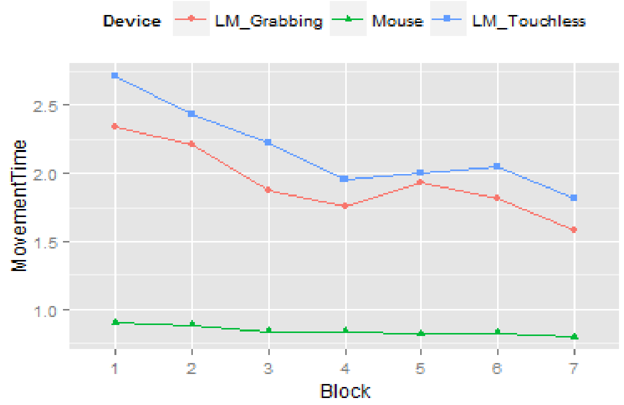

Figure 30: Movement Time for each block of each device (2nd experiment)

While this reduction is not drastic, this should somewhat increase the performance of the device. This was possible due to the way the click is performed. Contrary to the Touchless, the user may now position is finger more freely, as long as it is inside the target, since he/she will no longer be required to place it inside the Touching zone to perform the click. By simplifying this process, the selection task can be performed sooner.

Unfortunately, it is still not quite enough to match the computer mouse. These values have the same possible explanation as the last experiment. The users have little to no experience with the device and it is still being compared to one of the most frequent used devices whose experience in it is drastically higher.

It is no surprise that Throughput, as seen in **Figure 31**, has increased for the developed gesture but is yet unable to reach the computer mouse performance.

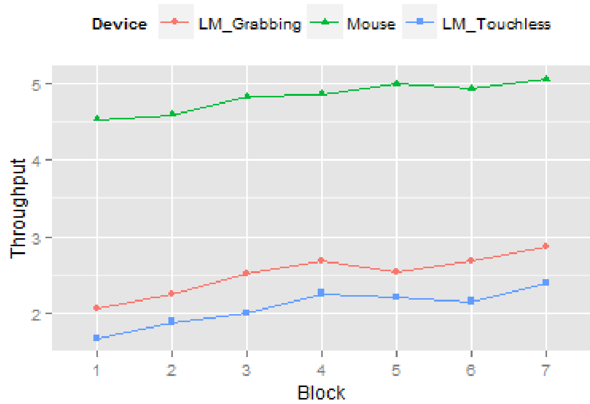

Figure 31: Throughput for each block of each device (2nd experiment)

When examining the Error Rate, which can be seen at **Figure 32**, it is possible to conclude that the new gesture has a higher error rate per block (ignore block 1 to 3) than the other options. This fact has been proven during the experiments and the cause is the motion required to perform the click. In this control mode, the user must open his/her hand, showing all the five fingers, to perform a click and close it, clasp it into a fist, to progress with the remaining task. The problem lies in the moment in-between. When tracking, and the user is in the process of showing/hiding his/her fingers, the Leap Motion sometimes detects more or less fingers then it should, only to capture or lose them again in the next frame. This leads to one or more unintended clicks being performed, resulting in many errors and the respective Error Rate. Perhaps, in future versions of the Leap Motion SDK and API, these tracking problems are solved. This would result in a decrease of the error rate of the developed control method.

Once again, the computer mouse is the device with the lowest Error Rate.

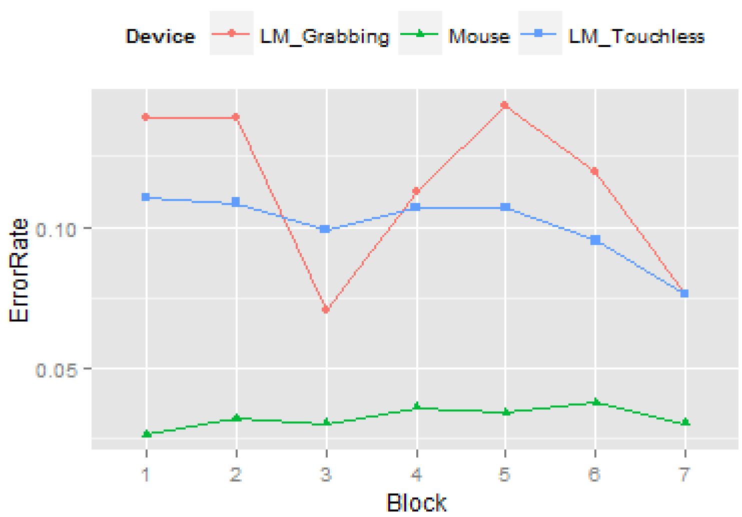

Figure 32: Error Rate for each block of each device (2nd experiment)

The result of seven accuracy measures can be found in **Figure 33**. When compared to the first experiment several no new conclusions can be reach in regards to the Leap Motion.

When comparing parameters like TRE, MDC, ODC, MV, ME and MO no conclusions can be taken as the confidence interval overlaps in every one of them.

However, there are two exceptions, and those are TAC and MDC. Regarding TAC, the new gesture reduced the number of times the pointer crosses the task axis when compared to the Touchless of either experiment. This might be due to the new way the pointer is now controlled, as the finger can be moved in a similar fashion to the mouse, allowing it to have a behavior similar to the computer mouse.

In terms of the MDC (whose confidence interval differed only 0.0579 between the Touchless and the Grabbing gesture), the new gesture was also able to reduce the number of times the pointer altered its trajectory from up or down. The similarity to the mouse behavior is again presented as an explanation, keeping in mind that the natural human tremor in the hand might be reason the reduction between both control methods in the Leap Motion was less than expected.

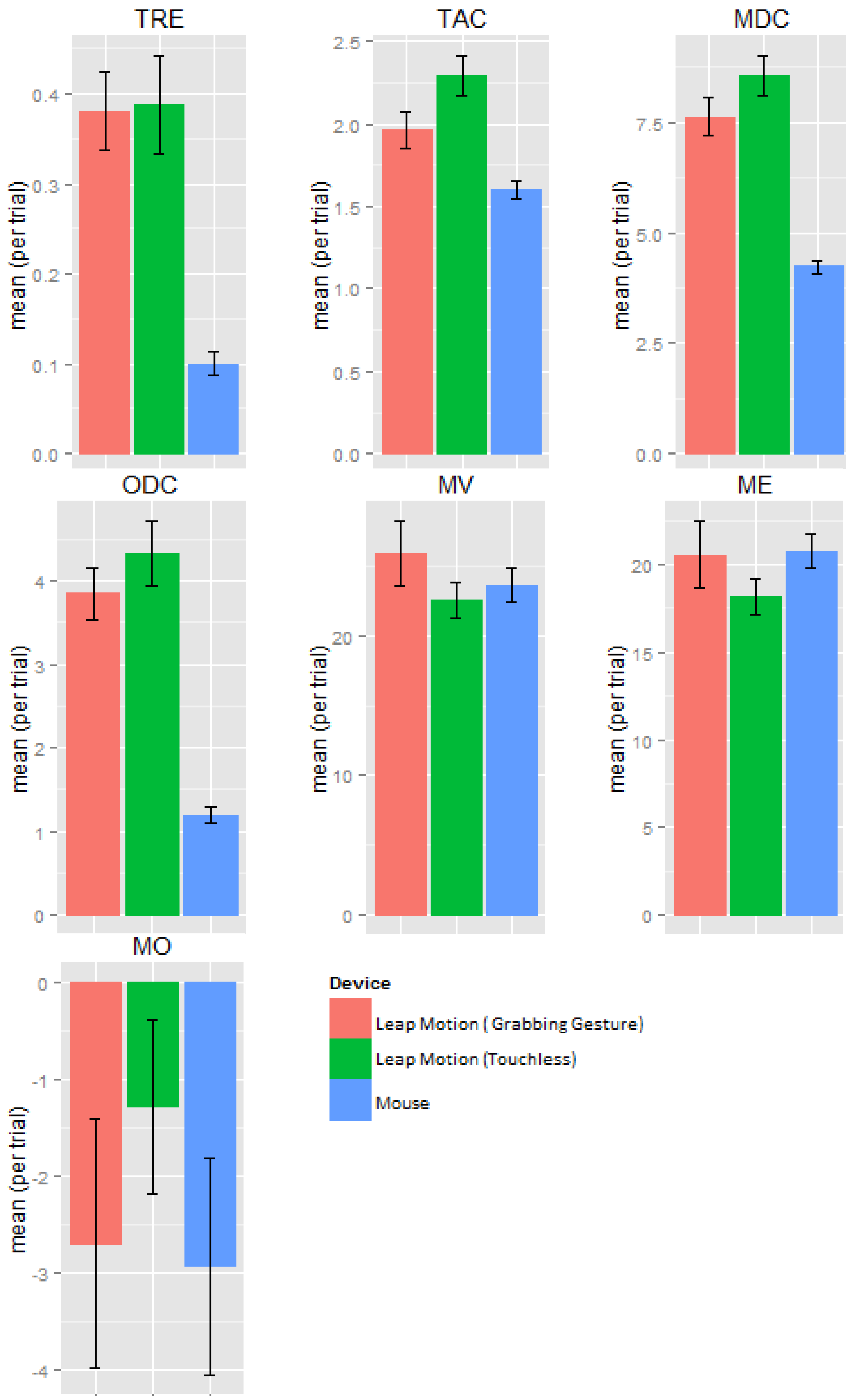

Figure 33: Seven accuracy measures for each device (2nd experiment)

When comparing the results of this experiment with the first one, several elements can be highlighted, namely in the TRE, MV, ME and MO. As the last three parameters are closely related the alteration in one easily could have affected the other.

Observing the Touchless, its TRE has slightly increase. The cause of such might be due to the number of users who participated in the experiment. This time all the users were first-time users, while in the first some of them were already familiarized with the device. The MV as also increased. The reason for this probably remains the same as the last parameter. The same can be said about ME.

Regarding the computer mouse, several accuracy measures have increased namely, TRE TAC, MV, ME and MO. In this experiment every user performed the computer mouse evaluation with the support of a mouse pad. However, several statistical tests were required to understand if this is indeed the influence of mouse pad or the work of chance.

In conclusion, the addition of a new gesture didn't bring any significant improvements in quantitative terms. The new gesture might perform relatively faster, which results in a better performance, but it also will increase the number of errors, which will frustrate the user and led him to dislike the device. Improvements in the detection and tracking of fingers either clasped together or clearly visible would lead to a reduction of the Error Rate and perhaps to an overall improvement.

5.2 Qualitative analyses

5.2.1 Evaluating the Leap Motion, touchpad and computer mouse

The data collected from the surveys performed during the first experiment is presented in the graphics contained in **Figure 34**, with a confidence interval of 95%.

The values presented for each graphic are the mean of the users' opinions when questioned about a specific topic. The scale of possible answers varies between 1 (most negative answer) to 5 (most positive answer). For example, when asked about the perceived accuracy of a device, users answered that the computer mouse was the most precise and the Leap Motion (using Touchless) the least precise.

All the values presented in the box plots can be consulted in the tables in **Survey results**, specifically sections **C1**, **C2** and **C3**.

The participants of the experiment were asked about their opinion and perceived value, at the end of each device evaluation, in several parameters, being these: force and effort required for operation, smoothness during operation, accuracy, operation speed, general comfort, overall operation of the input device and fatigue in several parts of the users body.

The results of the survey can be seen in **Figure 34**.

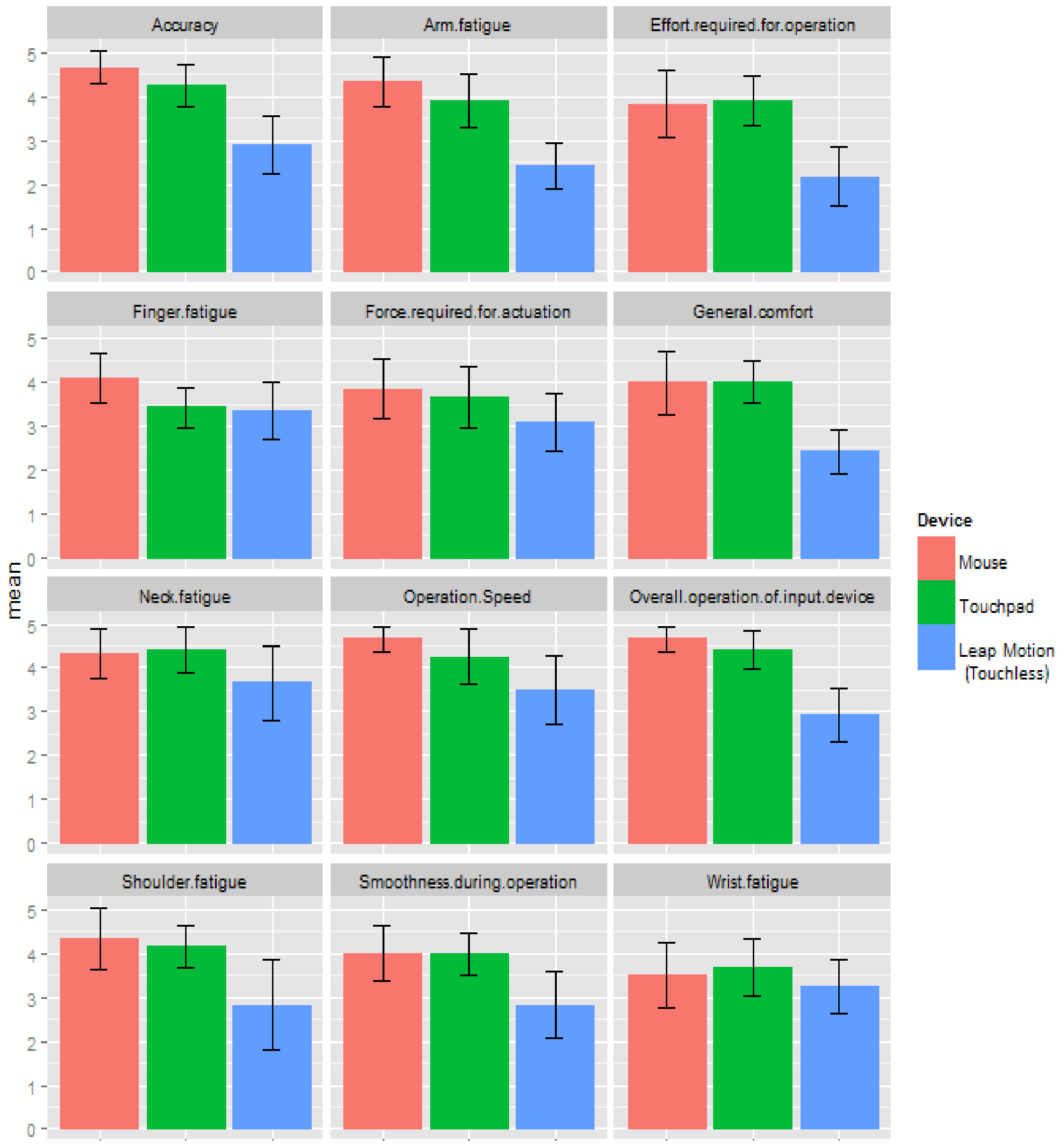

Figure 34: User opinions and perceived values during the first experiment

Remembering, to every question there was a scale of possible answers, ranging from 1 (very negative feedback) to 5 (very positive feedback).

By analyzing the data, it's possible to detect that confidence intervals between the devices overlaps frequently. Due to this reason, it is impossible to extract any precise conclusion in certain questions. For example, to every question, the computer mouse and touchpad confidence interval overlapped, being impossible to determine who was more precise, comfortable or caused less fatigue.

The same problem also happens to the Leap Motion. However, there are five exceptions allowing for conclusions to be taken.

The first exception is “accuracy”. Based in the users' opinions the Leap Motion was the less precise device. This result might be justified due to the clicking method used by the Touchless, which demanded the user to penetrate the Touching zone with their finger in horizontal position, otherwise, when getting close to the Touch Zone, the pointer would move and, sometimes leave the target. This was a source of frustration for the users in the first trials, when still learning to control the device. When starting to get tired, the user would have difficulty keeping the finger position as demanded by the control method which would result in the same consequences, lack of precision and frustration/indifference to the device.

The other exception is “arm fatigue”, which was considered one of the most demanding when compared to the other devices. The reason for this was easily detectable during the experiment and was due to the effort required by the users to maintain their dominant upper limb, who was responsible for controlling the pointer movement and perform a click, suspended in the air. With the arm and hand in the air, the user had more control over the device than, for example, when supporting his/her elbows in the table. Due to this, the user would have to exercise a lot of effort to remain with his/her limb in the air, resulting in an affliction known as "Gorilla's arm", and, not far off, pain. "Gorilla's arm" is an affliction identified by the human-computer interaction researchers in the early days when using touchscreens, in which one's arm feels heavy after waving it about for too long, as stated by MacGregor in [(Campbell 2013)].

The next two exception, “effort required for operation” and “general comfort”, can be tied to the last one. Users have find the Leap Motion one of the most effort requiring device when compared to the other two. A possible reason could be the arm fatigue and pain felt by the users. If the fatigue or pain was high and perceivable, it is an indication that a high amount of effort was being performed to control the device. Pain and discomfort are the opposite of comfort, whence one possible reason for the result.

The last exception is “overall operation of the input device”, which is lower (worse) than the computer mouse and touchpad. One possible explanation for this is the method the Touchless uses to perform a click. The user must place is finger inside the Touching zone in order for a click to take place and then remove it from the said zone. One of the first problems the user faces is to know when enough is enough. If the users goes to deep, a false click may happen or the device could lose the track of the finger controlling the pointer, looking for another in the process, making the pointer suddenly move somewhere and leaving the user confused. When removing the hand of the zone, if done to brusquely, the user would lose the visual feedback on how distant from the zone his/her finger was, requiring some time to understand the actual position. Sometimes the finger would leave the detection volume of the device, leaving the user confused why the pointer was not moving. All these reason could have lead the user to dislike the overall works of the Leap Motion.

Finally, the users were also asked which device they liked and disliked most. Of the twelve participants, eleven preferred the computer mouse and one preferred the touchpad, due to using it more frequently. Almost identical, eleven users disliked the Leap Motion and one user disliked the touchpad, claiming to distaste how slow and tedious the touchpad was when compared to the other devices.

In conclusion, from the perspective of the users who participated in the experiment, none showed interest in using the Leap Motion in a day-to-day basis to control its computer due to how strenuous the device is. Also, with a lower performance than the computer mouse or touchpad, the Leap Motion does not appear to be a viable alternative to use in high demanding and production tasks.

5.2.2 Evaluating the Leap Motion with different control modes

The second experiment was performed in order to determine if by changing the control mode, the users would be more apt and prefer to use the Leap Motion. The control mode evaluated was the one developed within this dissertation and the Touchless. The computer mouse was again evaluated.

The analysis was performed very similarly to the last section being the data collected from the surveys present in the box plots, using once again a confidence interval of 95%. The questions and possible answers were also the same with a little exception on the last survey, where some extra queries where performed in order to try to identify which gesture/control mode the participants preferred.

All the values present in the box plots, in **Figure 35**, can be consulted in the tables in addendum to this document, specifically **Survey results**, section **C4**, **C5** and **C6**.

Unfortunately, based on the confidence interval, there are not many conclusions that can be taken due to the overlapping between both control modes of the device. While there are some positive increases when using the Grabbing gesture this results could be the work of chance.

However, one point should be highlighted. In overall, the computer mouse continues to be seen as the best device for the participants. Independent of the question, the mouse has always the best score and, in the worst case scenario, it never is exceed by the Leap Motion controller. In the question where participants were asked whose device caused less neck fatigue the mouse was voted by all as having cause no fatigue.

When asked about the device preference, whose options were only two this time, all nine participants of the experiment declared that the computer mouse had been their favorite device, which as expected based on the commentaries from the last experiment and the results of the current one.

The users were also asked of the two control modes which they preferred when taking into account certain parameters. When asked which had seem more comfortable to use, seven users answered that the Grabbing gesture was their favorite one. The justification given is that, when compared to the Touchless, less effort is required to be precise and to perform a click, having more freedom of movement. One of the participants also said that this control mode seemed to resemble the way the computer mouse was controlled.

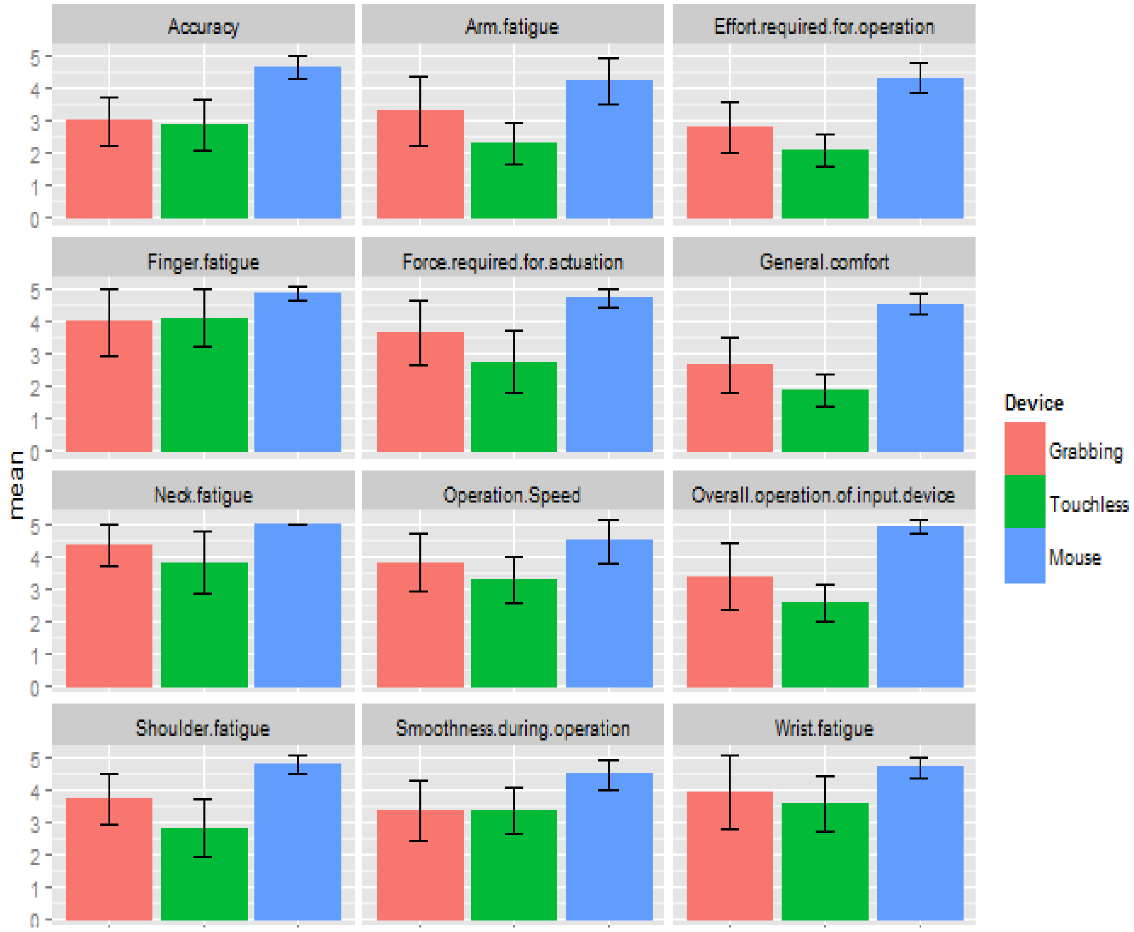

Figure 35: User opinions and perceived values during the second experiment

When questioned about which method would was more intuitive six users answered that the developed gestured seemed to be intuitive, while three said Touchless. One of the users who preferred the Touchless said that using two hands is more troublesome than using just one, who alone could perform the same tasks with no problem.

In terms of accuracy, the users were more divided. Five claimed that the “Grabbing gesture” was the most precise while four elected Touchless. Based on this, it will be dangerous to claim that one of the gesture is more precise than the other and, unfortunately, the doubt will remain.

Finally, when inquired about which control method the user preferred, eight in nine users preferred the developed gesture.

In conclusion, implementing the new gesture could possible bring new improvements but such could not yet be proven. By performing this experiment again with extra participants could prove fruitful. Also, a more profound statistical analyses could help bring certain conclusions that simple analyses could not show.

5.3 Posture and comment analyses

5.3.1 Evaluating the Leap Motion, touchpad and computer mouse

In terms of the devices, the Leap Motion was the device where more feedback was given. The other devices receive less commentary from the participants perhaps from being more commonly used.

After an intensive use, the Touchpad received no complaints in terms of pain or discomfort, while some users cited it as less productive than the mouse, making the experience somewhat tedious, and more repetitive than the Leap Motion, since the gesture, while being the same, could be adapted or performed in a slight different way.

On the contrary, after intensive use the mouse received some complaints about discomfort and tension in the zone of the wrist and fingers. The cause of this probably of when using the mouse, users don't move their forearms, being the movement of the computer mouse at the responsibility of the wrist who is performing all the movements necessary to move the device. The pain felt in the hand/fingers as probably due to the fast clicking process that some users were executing because they were confident in the usage of the device. One user felt discomfort in the upper part of the arm. Also some users complained that using the mouse lead to produce of a lot of sweat.

Some users also complained about the speed definitions of the device and of the computer mouse used (they were too slow).

In terms of the Leap Motion, the common position taken by the users was backs against the chair, some slightly curved to the front, others straight. One finger of the dominant hand was responsible for controlling the pointer movement through the *Touchless* application. Most of the users used their index finger while one user used his/her thumb. The other fingers were hidden in order to not to activate other functions of the application (like the scroll function). Some users would also leave the thumb showing which sometimes caused the software to perform unintended clicks when the hand went too deep the touch zone.

The auxiliary hand, being unneeded, would remain quiescent on the table, over the user legs or suspend at the limit of the chair's arm, with the respective forearm laying on it. Frequently, the user would support his/her auxiliary hand's elbow in the table and support their head or mouth in the respective hand.

Experimental Results

One of the problems detected in terms of posture is that, very frequently, users had to keep their dominant upper limb suspend in the air in order to execute the motions and have control over the device. This, after a small time, caused the user to start feeling discomfort and the necessity to make a pause. Even with pauses (the user was allowed to make a pause at the end of every trial/sequence for how long he/she wished) this discomfort would return and often ended in pain. In the end, those who perform the evaluation with their dominant upper limb suspended in air, no matter how many pauses were taken, felt pain in their arm. The main source of this problem is effort required to keep the limb in the air. This phenomenon was already known and expected and is called the "Gorilla arm" due to how heavy the arm will start to get over time due to waving it around for too long. Sometimes this discomfort/pain would extended to the shoulders, neck and back.

In terms of fingers, one of the other problems found was that after some time keeping the finger under tension, necessary for keeping the finger extended in order to control the device, stress and fatigue would appear. If the evaluation proceeded, this stress would end up in and pain. The fatigue however, even after a short pause, would not go away and, as a natural response, the finger would started to relax. This made the user lose some precision. When the finger approached the touch zone to perform a click, the pointer would sometimes move outside the target area and the click would end up wrong.

Tension was also required when trying to point very precisely at a point and forcing the hand to remain motionless. Long periods of tension would result in the same consequences. One other aspect related to the fingers is that most of them (more than three) must remain hidden otherwise a different function of the *Touchless* would be activated (one example is the scroll function). Keeping the fingers hidden is another task demanding the user efforts.

Some users would prefer to leave their thumb in a relaxed position (hiding it would cause some discomfort and pain after a while). While acceptable, it could be one more cause of a false click. Several users went to deep in the touching zone when performing a click. This could cause the thumb to also perform a click. If after a successful click another click has performed, this click would make the selection an unsuccessful selection since the target would have already changed. Also one of the problems of diving too far in the touch zone is that the device might lose the tracking of the controlling finger. When this happened, the device would look for another finger. This made the pointer onscreen jump to a different position, sometimes more than once.

One of the users performed high performance sport and joked that the Leap Motion evaluation seemed like one of the exercises he/she used to do. This user was one fastest to perform the evaluation due to his/her physical conditions.

During the experiment, some users would attempt to reposition the Leap Motion closer or farther from them in an attempt to place their elbows on the table and achieve some comfort. This lead to a reduction in the performance (user took long to perform a successful click) and an increase in the number of times the device lost tracking. This can be explained with trajectory performed by the finger when the elbow is static. When the finger tries to move forward it starts describing an arc, as it cannot go forward, with the radius of the forearm and the elbow as the center. Unconsciously the user doesn't adapt the finger ending up with it pointing straight at the device (parallel to the invisible plane where the click occurs). In this situation, the device is sometimes enable to track the finger and the finger position is lost.

Several users also commented that if the Leap Motion was in lower position , the elbows could be supported on the table and the hands wouldn't have to go so higher (being less tiring) or so low (leaving the detection area). Some users also tried to use the table borders as a support point. The usefulness of this tactic depended on the user complexion.

In terms of selecting targets, users remarked that the lower targets were the hardest to select. One of the causes of this problems could be the Leap Motion tracking volume, which is an inverse pyramid. In other words, the closer an object/finger is to the device, the smaller is detection range is. When moving up-or-down, users would unintentionally get their controlling finger closer or farther from the invisible plane, which was a common event in the first trials, leading to wrong clicks and frustration. Moving from left-to-right or right-to-left as easier to do.

Fortunately, there was a user who managed to complete all the trials with Leap Motion without feeling pain, leading to believe that the position and control he/she used is the correct. The user supported his/her elbows on the table and used his/her thumb to move the pointer and perform the clicks (the index finger was hidden). Due to the user constitution, he/she was able to find a comfortable position and able to reach every target without rising his/her elbows. The downside is that the finger tracking failed several times.

Finally, users with certain functional inabilities, like tendinitis, or with back problems, due to incorrect sitting postures or age, had greater difficulty performing the evaluation and were unable to complete it. This leads to believe that the device is not recommended to be used but persons with the said problems.

5.3.2 Evaluating the Leap Motion with different control modes

When evaluating the postures from the second experiment, no new information appeared that haven't been captured in the last experiment relatively to the users postures when using the computer mouse and Touchless. As such, in order to avoid repeating information, only the new developed gesture postures will be analyzed.

The common posture taken by the users when using this control method is to place his/her back against the chair. Some users kept their back straight while others had it slightly curved, perhaps due to the way he/she uses the computer in a day-to-day basis.

The dominant hand, whose finger, normally the index, is responsible for controlling the pointer movement, was placed above the table, suspended in the air. Other users would support their elbows in the table, to avoid keeping their arms in the air, and use the forearm as an axis for the movement, moving the hand only whenever needed. Some users, depending on his/her physique, opted for supporting their forearms at the table borders.

The auxiliary hand, the hand responsible for performing a click, would have a similar position to the dominant hand. In other words suspended in the air above the device or with its elbows supported in the table. In any of the two, the auxiliary hand had to be inside the device detection volume. One interesting point perceived was that the higher the user would raise his/her hand, the faster he/she would start feeling tired but the detection of the device seemed better in these moments. A user declared that having the hands suspended in the air has not the right position due to how much effort that required.

When not satisfied with his/her position, the user would try to change it until a comfortable position was achieved.

In terms of comfort, as expected, the developed gesture also caused some strain and discomfort. However, contrary to the Touchless in both experiments (in the second experience one user was unable to finish the evaluation using the Touchless), all the users were able to finish the evaluation of the Leap Motion with this control mode. Also, every user was able to find a somewhat comfortable position to perform the experiment. A possible reason for this was because the clicking process had been simplified, now the user could place their hands above the device with more freedom because the user no longer had to worry about entering (or exiting) the Touching zone, which occupied half of the device detection volume.

A user declared feeling some discomfort in the back and in the dominant hand. One possible reason for the first one was due to controlling the device with the hands suspended in the air. For the second, it could have been due to one, or more, abrupt movements performed with the wrist or fingers.

One other user was able to perform the evaluation controlling the mouse with the Thumb instead of the index finger. The user affirmed that this was less strenuous but, sometimes, had some difficulties being detected.

In terms of the gesture, some issues were found in the hands detection. Sometimes, due to the posture taken by the user, the dominant hand would leave the device detection volume, making the pointer jump to the position of the auxiliary hand, which would frustrate and confuse the user.

This is a technical issue of the developed gesture. The control mode was programmed that, when the dominant hand was lost and one hand remained, the remaining hand would be the new dominant hand.

In terms of detection, several issues and click errors were detected when hands were too close to each other. This, however, was already an expected problem and was due to this that the GUI had to be translated somewhat to the right of the screen center, in order to avoid the hands getting close or overlapping each other.

In conclusion, the new gesture seemed to have been accepted by the users as an alternative to the Touchless, being somewhat less painful but nevertheless strenuous. If the experiment proceed behind the specified number of trials, the user would end up in the same pain and discomfort as if using the Touchless. Due to this, the Leap Motion controller is not a device recommended for these type of tasks.

5.4 Possible guidelines when using the Leap Motion

The following points are recommendations to the users or developers trying to use the Leap Motion in WIMP based interfaces or very similar 2D interfaces.

1. The Leap Motion, while not recommended, can be used as substitute of a input device used to interact with a computer via a WIMP interface in selection tasks;
2. The Leap Motion is not recommended for high performance tasks. The controller is slower than many input devices and demands a high effort to be used.
3. The Leap Motion, while able to interact with WIMP interfaces, should not be used intensively. Frequent pauses are recommend, otherwise discomfort and pain might be felt.
4. The Leap Motion is not recommended to be used with persons with physical problems / limitations on the back or upper limbs, as the device demands a certain effort from these body parts.
5. When using the Leap Motion a comfortable position should be taken. This position might vary from user to user but keeping the hands suspended in the air should be avoided as this is one of the causes of the discomfort and pain felt when using the device. A common comfort position is supporting one or both elbows in a surface.
6. When programming a gesture or interface compatible with the Leap Motion, close attention should be given to the movements and the detection volume of the device so

not to make users perform hurtful motions or prevent them from positioning themselves in a comfortable posture.

7. Gesture or motions that use the Leap Motion's Touch Zone functionality should be take caution not to disrespect point 6;
8. When drawing an interface, any element that requires selection (for example, menus, icons, and buttons) should be placed in the upper part of the screen. Elements placed in the lower screen are difficult to select;
9. If using the Leap Motion's Touch Zone functionality, elements that require selection should be placed in a line (left-to-right, or vice-versa) and not in a column (up-to-down, or vice-versa). When a user moves his/her hand, the probability of entering the Touch Zone unintentionally is higher when moving up or down than when moving left or right;
10. While it is not always possible, the Leap Motion controller should be placed somewhat below the user. This allows the user to place his /her hands more easily in the upper part of the device detection volume, allowing for a larger area of movement. This also demands less effort from the user to reach this area since he/she does not need to elevate the arms so high;
11. The Leap Motion controller is not recommended for high precision tasks. While it is possible to use it to perform these tasks, the effort required to maintain the position and perform a precise movement demands effort and causes discomfort to the user;
12. The width of the buttons or other targets should be given special attention:
    - The larger the width of the target the easier is to select it, the less effort the user is required to perform;
    - The Target Re-entry of the device is relatively high. Buttons should be large enough to avoid the pointer leaving unintentionally. Also the time menus remain open when the pointer leaves them unintentionally should also be higher than the one currently used for the computer mouse or touchpad;
13. A pointer being controlled by the Leap Motion controller has a higher tendency of naturally moving up/down than left/right.
14. The movement of the pointer being controlled by the Leap Motion, at least when using the application Touchless, is very similar to a pointer whose movement is being controlled by a computer mouse or a touchpad;

These points are not rules and are not meant to be followed blindly but are conclusions taken after the evaluation of the Leap Motion controller using two different control modes. Also, as time passes, the device's SDK or API will certainly evolve, making it more simple and easy to use, and more people might start using it more frequently, leading to some of these points (if not all) becoming obsolete.

5.5 Summary and Conclusion

In this chapter all the data collected during the first and second experiment performed within this dissertation were presented. In the first experiment the computer mouse, the touchpad and the Leap Motion controller were evaluated, either quantitatively, using the ISO 9241-9:2000 standard, MacKenzie, Kauppinen and Silfverberg accuracy measures and Error rate, either qualitatively, by performing surveys and analising the user postures and feedback giving during the experiment. For the second experiments, only two devices were evaluated, the computer mouse and the Leap Motion, but this time using two different control methods.

In the end, the Leap Motion has proven not to be adequate to high performance tasks due to how slow and demanding it is when performing the tasks. The participants who performed the experiment also said to prefer the computer mouse to the Leap Motion due to being more comfortable and fast. In terms of control modes, the users prefered the developed gesture, although it was not possible to prove how much benefits (if any) it brought in terms of usability.

Chapter 6

Conclusions and future work

6.1 Objetives satisfaction

This chapter marks the end of this document. The initial objective of this dissertation was to evaluate the Leap Motion using the seven accuracy measures, proposed by [(MacKenzie, Kauppinen and Silfverberg 2001)], and further complement it by determining its performance and error rate. The same evaluation was performed for the computer mouse and touchpad in order to compare the results between each other, determining where the Leap Motion could perform better or worse than the reamaining.

The information collected was analyzed and a conclusion as reached. The Leap Motion has less comfortable, precise and had a lower performance than the other devices. The users opinions also indicated that the Leap Motion would not be choosen as a viable alternative to control their computers.

With the main objective complete, it proved interesting to determine if the values collected earlier could be improved by using a new gesture / control mode. Several gestures were developed and experiemented with it. The “Grabbing” gesture, a new gesture developed within this dissertation, was choosen by the users in an informal session as the preffered from the six gestures available.

The same evaluation has performed using the new gesture. The last gesture and the computer mouse were used again. In the end, the results revealed a small improvements but not enough to garner more interest from the users.

With this all the objectives of this dissertation were achieved. The Leap Motion was able to interact with WIMP interfaces but was unable to be an adequate tool to be used in a day-to-day basis for controlling this type of interfaces.

For users or programmers who still wish to use this paradigm, a couple of guidelines were proposed in order to try to decrease the users efforts and discomfort when using the Leap Motion, hoping to create a more pleasent interaction.

6.2 Future Work

There is, however, several points that should be highlighted. First, the Leap Motion is still new device, giving its first steps. The computer mouse was not created as perfect as it is now and so won't the Leap Motion. With the passing of time, several improvements might appear, easing the burden caused on the users and transforming the device in a more robust tool. In fact, an upgrade for the API and SDK is already available for developers and, soon, for commercial use as well.

Second, contrary to the computer mouse or touchpad, the Leap Motion does not have accelaration. This function is what allows the pointer on screen to move so fast even though only a little movement was done, allowing for a comfortable and effortless interaction.

Based on this two points, several more experiements could be extracted, namely:

- Perform the same experiment with users that already have the same experience with the Leap Motion as they have with another input device like the computer mouse;
- As users might already be biased and resistant to changes, performing this same experiment with younger users, with little to no experience with other input devices might reveal a more interesting feedback and interest from the subjects;
- Applying acceleration to the device could also prove fruitful. Performing the same experiment with acceleration available and again compare the results with other devices who also possess the same functionality might prove insightful;

This would allow for extra conclusions to be taken and shed some light on the Leap Motion, determining if it is a tool to be used only for entertainment or if it is the start of something bigger and the beginning of a new form of interaction.

References

Autodesk. n.d. *Autodesk Main Webpage.* Accessed February 3, 2014. http://www.autodesk.com/.

Baig, Edward C., and Jon Swartz. 2013. "Loco motion: Out of touch, but in control." *USA TODAY*, March 11.

Campbell, MacGregor. 2013. "OK computer." *New Scientist*, 40-43.

Cheong, Kyeong-kyun, Insoo Kim, Sung-kwon Park, and Yong-jin Park. 2011. "User performance measures for evaluating interactive TV pointing devices." *IEEE Transactions on Consumer Electronics* 1236-1244.

Douglas, Sarah A., Arthur E. Kirkpatrick, and I. Scott MacKenzie. 1999. "Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard." *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit CHI 99* 215-222.

Garber, Lee. 2012. "Great Leap Forward in Gesture-Control Technology." *IEEE Computer Society*, July: 21-22.

Gardiner, Bryan. 2012. "Point to Click." *Popular Science*, December: 14.

Google. 2011. *Google Docs.* Acedido em 2014 de June de 19. http://www.google.com/google-d-s/intl/pt-PT/tour1.html.

International Organization for Standardization. 2007. "ISO 9241-400:2007." *Ergonomics of human--system interaction -- Part 400: Principles and requirements for physical input devices.*

International Organization for Standardization. 2000. "ISO 9241-9:2000." *Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 9: Requirements for non-keyboard input devices.*

Leap Motion. 2014. *Leap Motion Developers Portal.* Accessed May 29, 2014. https://developer.leapmotion.com/.

References

—. 2012. *Youtube - Introducing the Leap Motion.* May 21. Accessed February 2, 2014. http://www.youtube.com/watch?v=_d6KuiuteIA.

Leap Motion, Inc. n.d. *Leap Motion.* Accessed January 24, 2014. https://www.leapmotion.com/.

—. 2013. *Leap Motion Blog - Release Date Updated.* April 25. Accessed February 5, 2013. http://blog.leapmotion.com/post/48872742284/release-date-update.

—. 2013. *Leap Motion Blog - Surgeons navigate medical images in the air.* October 15. Accessed February 5, 2013. http://blog.leapmotion.com/post/64152846497/surgeons-navigate-medical-images-in-the-air.

MacKenzie, I. Scott, Abigail Sellen, and William A. S. Buxton. 1991. "A comparison of input devices in element pointing and dragging tasks." *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* 161-166.

MacKenzie, I. Scott, Tatu Kauppinen, and Miika Silfverberg. 2001. "Accuracy Measures for Evaluating Computer Pointing Devices." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 9-16. Seattle, Washington, USA: ACM Press.

MacKenzie, I., and Shaidah Jusoh. 2001. "An Evaluation of Two Input Devices for Remote Pointing." *Engineering for Human-Computer Interaction* 235 - 250.

Malcolm, John. 2013. *Leap Motion and Corel's Painter Freestyle - YouTube.* 21 de July. Acedido em 6 de February de 2014. http://www.youtube.com/watch?v=0FEeYtpALkM.

Markowitz, Eric. 2013. "Companies You Can't Afford to Miss at SXSW." *Inc.*, March: 17-18.

—. 2013. "Wave Your Hand, Control Your Computer. ." *Inc.*, June: 112-113.

Marks, Paul. 2012. "Wave away your mouse." *New Scientist*, December 22.

McArthur, Victoria, Steven J. Castellucci, and I. Scott MacKenzie. 2009. "An empirical comparison of "wiimote" gun attachments for pointing tasks." *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems - EICS '09* 203-208.

Microsoft Corporation. 2013. *Microsoft Kinect Features.* Accessed January 24, 2014. http://www.microsoft.com/en-us/kinectforwindows/discover/features.aspx.

Nabiyouni, Mahdi, Bierswar Laha, e Doug A. Bowman. s.d. *Poster: Designing Effective Travel Techniques with Bare-hand Interaction.* Center for Human-Computer Interaction and Department of Computer Science, Virginia Tech.

Natapov, Daniel, and I. Scott MacKenzie. 2010. "The trackball controller: improving the analog stick." *Proceedings of the International Academic Conference on the Future of Game Design and Technology* 175-182.

Natapov, Daniel, Steven J. Castellucci, and I. Scott MacKenzie. 2009. "ISO 9241-9 evaluation of video game controllers." *Proceedings of Graphics* 223-230.

New Scientist. 2013. “One per Cent .” *New Scientist*, 9 de March: 25-25.

Nintendo of America, Inc. n.d. *Nintendo Wii Console Specs.* Accessed January 24, 2014. http://www.nintendo.com/wii/what-is-wii/#/tech-specs.

References

Okada, H., and T. Akiba. 2010. "Fitts' Law Index of Difficulty Evaluated and Extended for Screen Size Variations." In *User Interfaces*, by InTech, 229-238. InTech.

Pogue, David. 2013. *The New York Times - Personal Tech.* July 24. Accessed February 2, 2014. http://www.nytimes.com/2013/07/25/technology/personaltech/no-keyboard-and-now-no-touch-screen-either.html?pagewanted=1&_r=0&partner=rss&emc=rss.

Processing Fondation. n.d. *Processing 2.0.* Accessed June 17, 2014. http://www.processing.org/.

R-Project. n.d. *The R Project for Statistical Computing.* Accessed June 9, 2014. http://www.r-project.org/index.html.

Savage, Neil. 2013. "More than a Mouse." *communications of the acm*, November: 15-16.

Silva, Eduardo S., Jader Anderson O. de Almeida Abreu, Janiel Henrique P. de, Veronica Teichrieb, e Geber L. Ramalho. 2013. *A Preliminary Evaluation of the Leap Motion Sensor as Controller of New Digital Musical Instruments.* Recife, Brasil: Centro de Informática – Universidade Federal de Pernambuco (UFPE).

Sutton, Jeremy. 2013. "Air painting with Corel Painter Freestyle and the leap motion controller: a revolutionary new way to paint!" *ACM SIGGRAPH 2013 Studio Talks.*

Thomas, Bruce H. 2006. "Evaluation of three input techniques for selection and annotation of physical objects through an augmented reality view." *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality* 33-36.

Vikram, Sharad, Lei Li, and Stuart Russell. 2013. "Handwriting and Gestures in the Air, Recognizing on the Fly." *CHI 2013 Extended Abstracts.*

Weichert, Frank, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. "Analysis of the Accuracy and Robustness of the Leap Motion Controller." *Sensors* 6380-6393.

Zhao, Haixia. 2002. *Fitts' Law: Modeling Movement Time in HCI.* October. Acedido em 10 de February de 2014. http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/fitts.html .

Zucco, Joanne .E., Bruce H. Thomas, and K. Grimmer. 2006. "Evaluation of Four Wearable Computer Pointing Devices for Drag and Drop Tasks when Stationary and Walking." *2006 10th IEEE International Symposium on Wearable Computers.*

Zucco, Joanne E., Bruce H. Thomas, Karen Grimmer-Somers, and Andy Cockburn. 2009. "A Comparison of Menu Configurations and Pointing Devices for Use with Wearable Computers while Mobile and Stationary." *2009 International Symposium on Wearable Computers* 63-70.

Appendix A

Questionnaires

A1: Demographic and Computer Literacy survey

Device evaluation / Avaliação de Dispositivos

The following questionnaire intends to be a follow up of the evaluation perfumed by the user. The presented questions will be related to the experience using the three devices (Leap Motion, Mouse and Touchpad) and some demographic questions.

O seguinte questionário apresenta-se na sequência da avaliação realizada anteriormente pelo utilizador. As questões aqui colocadas visão perceber a sua experiência ao utilizar essas tecnologias (Leap Motion, rato e painel táctil), nomeadamente em termos de conforto e usabilidade bem como algumas perguntas demográficas.

* Required

Gender / Género: *

- Male / Masculino
- Female / Feminino

Age / Idade *

- [0 - 14]
- [15 - 20]
- [21 - 25]
- [26 - 30]
- [31 - 35]
- [36 - 40]
- [41 - 45]
- [46 - 50]
- [51 - 55]
- [56 - 60]
- 60+

Appendix

How often do you use a computer? / Com que frequência utiliza o computador? *

- Never / Nunca
- Seldom / Raramente
- Often / Frequentemente
- Everyday / Todos os dias

How often do you use a computer mouse? / Com que frequência utiliza um rato de computador? *

- Never / Nunca
- Seldom / Raramente
- Often / Frequentemente
- Everyday / Todos os dias

How often do you use a Touchpad? / Com que frequência utiliza um paínel táctil? *

- Never / Nunca
- Seldom / Raramente
- Often / Frequentemente
- Everyday / Todos os dias.

How often do you use a Leap Motion? / Com que frequência utiliza o Leap Motion? *

- This was my first time. / Esta foi a minha primeira vez
- Seldom / Raramente
- Often / Frequentemente
- Everyday / Todos os dais.

A2: Individual device evaluation survey

Individual device evaluation / Avaliação individual dos dispositivos

The following questionnaire intends to be a follow up of the evaluation performed by the user when using the Leap Motion device.
O seguinte questionário apresenta-se na sequência da avaliação realizada anteriormente pelo utilizador utilizando o Leap Motion.

* Required

Leap Motion / Touchpad / Computer Mouse

Force required for actuation / Força necessária para o funcionamento: *
The force you had to apply to make the device move. / A força que teve que aplicar para fazer o dispositivo mexer-se.

	1	2	3	4	5	
Very uncomfortable / Muito desconfortável	○	○	○	○	○	Very comfortable / Muito confortável

Smoothness during operation / Suavidade durante o funcionamento: *
Did the device move without difficulty? / O dispositivo deslocava-se sem dificuldade?

	1	2	3	4	5	
Very rough / Muito brusco	○	○	○	○	○	Very smooth / Muito suave

Appendix

Effort required for operation / Esforço necessário para o funcionamento: *
How much effort was required to keep the device running? / Foi necessário muita esforço para manter o dispositivo em funcionamento?

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

Accuracy / Precisão: *

Very inaccurate / Muito impreciso	1 ○	2 ○	3 ○	4 ○	5 ○	Very accurate / Muito preciso

Operation Speed / Velocidade de funcionamento: *

Unacceptable / Inaceitável	1 ○	2 ○	3 ○	4 ○	5 ○	Acceptable / Aceitável

General comfort / Comforto geral: *

Very uncomfortable / Muito desconfortável	1 ○	2 ○	3 ○	4 ○	5 ○	Very comfortable / Muito confortável

Overall operation of input device / Funcionamento global do dispositivo de entrada: *

Very difficult (to use) / Muito difícil (de usar)	1 ○	2 ○	3 ○	4 ○	5 ○	Very easy (to use) / Muito fácil (de usar)

Finger fatigue / Cansaço nos dedos: *

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

Wrist fatigue / Cansaço nos pulsos: *

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

Arm fatigue / Cansaço nos braços: *

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

Shoulder fatigue / Cansaço nos ombros: *

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

Neck fatigue / Cansaço no pescoço: *

Very High / Muito alto	1 ○	2 ○	3 ○	4 ○	5 ○	Very low / Muito baixo

A3: Favorite device survey (1st experiment)

Input device preference / Dispositivo de entrada preferido

* Required

Which was your favorite device? / Qual o dispositivo que mais gostou? *

- Leap Motion
- Touchpad
- Computer Mouse

Which device did you prefer less? / Qual o dispositivo que menos gostou? *

- Leap Motion
- Touchpad
- Computer Mouse

A4: Favorite device and gesture survey (2nd experiment)

Input device preference / Dispositivo de entrada preferido

* Required

Which was your favorite device? / Qual o dispositivo que mais gostou? *

- Leap Motion
- Computer Mouse

Of the used gestures, which one did you find more intuitive? / Dos gestos que utilizou, qual achou o mais intuítivo? *

- Touchless
- Grabbing / Agarrar

Of the used gestures, which one did you find more comfortable? / Dos gestos que utilizou, qual achou o mais confortável? *

- Touchless
- Grabbing / Agarrar

Of the used gestures, which one did you find more accurate? / Dos gestos que utilizou, qual achou o mais preciso? *

- Touchless
- Grabbing / Agarrar

Of the used gestures, which one did you liked most? / Dos gestos que utilizou, qual o que mais gostou? *

- Touchless
- Grabbing / Agarrar

Appendix B

Evaluation results

B1: - MacKenzie, Kauppinen and Silverberg's accuracy metrics (1st exp.)

	Target Re-entry		Task Axis Crossing		Movement Direction Change		Orthogonal Direction Change	
Device	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Mouse	0,1035	0,0880	1,6640	0,3708	4,8563	0,9564	1,1896	0,5855
Touchpad	0,1437	0,1175	1,2563	0,3167	4,4760	1,1494	1,0429	0,5521
Leap Motion (Touchless)	0,3512	0,2083	2,3523	0,6233	8,7760	2,4333	4,4984	2,1849

	Movement Variability		Movement Error		Movement Offset	
Device	Mean	SD	Mean	SD	Mean	SD
Mouse	20,5223	6,8842	18,4082	5,2748	-1,6771	5,9268
Touchpad	20,0459	6,8106	16,5477	5,2440	-1,5834	6,3551
Leap Motion (Touchless)	20,4654	6,2380	16,2696	5,3271	-1,4923	4,5889

B2: Throughput and Error Rate (1st experiment)

	Throughput		Movement Time	
Device	Mean	SD	Mean	SD
Mouse	4,8479	0,4315	0,8677	0,0902
Touchpad	3,2148	0,4219	1,3107	0,1998
Leap Motion (Touchless)	2,1619	0,3950	2,1071	0,5165

	Error Rate	
Device	Mean	SD
Mouse	0,0187	0,0384
Touchpad	0,0213	0,0452
Leap Motion (Touchless)	0,0672	0,0765

Appendix

B3: MacKenzie, Kauppinen and Silverberg's accuracy metrics (2nd exp.)

	Target Re-entry		Task Axis Crossing		Movement Direction Change		Orthogonal Direction Change	
Device	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Mouse	0,1000	0,0790	1,6024	0,3403	4,2300	0,8670	1,1952	0,5569
Leap Motion (Created gesture)	0,3805	0,2625	1,9590	0,6659	7,6186	2,5817	3,8410	1,8586
Leap Motion (Touchless)	0,3871	0,3299	2,2938	0,7065	8,5514	2,7003	4,3205	2,3252

	Movement		Movement Error		Movement Offset	
Device	Mean	SD	Mean	SD	Mean	SD
Mouse	23,5688	7,4618	20,7355	5,8159	-2,9390	6,8214
Leap Motion (Created gesture)	25,8543	13,8305	20,5660	11,3128	-2,7062	7,7857
Leap Motion (Touchless)	22,5627	7,6591	18,1455	6,0165	-1,2836	5,4727

B4: Throughput and Error Rate (2nd experiment)

	Throughput		Movement	
Device	Mean	SD	Mean	SD
Mouse	4,9522	0,6471	0,8220	0,1112
Leap Motion (Created gesture)	2,6877	0,5519	1,7727	0,5825
Leap Motion (Touchless)	2,2509	0,3602	1,9543	0,4340

	Error Rate	
Device	Mean	SD
Mouse	0,0348	0,0449
Leap Motion (Created gesture)	0,1129	0,1272
Leap Motion (Touchless)	0,0962	0,1009

Appendix C

2 Survey results

C1: Leap Motion (Touchless) evaluation (1st experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort	Overall operation of input device
Average	3,0833	2,8333	2,1667	2,9167	3,5	2,4167	2,9167
Standart Deviation	1,1645	1,3371	1,1934	1,1645	1,3817	0,9003	1,0836
Meaning	Not comfortable but not uncomfortabl	Not smooth and not rough	High	Not accurate but not innacurate	Almost Acceptabl e	Uncomfortable	Not difficult but not easy

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Average	3,3333	3,25	2,4167	2,8333	3,6667
Standart Deviation	1,1547	1,0553	0,9003	1,8007	1,4975
4 Meaning	Not high but not low	Not high but not low	High	Not high but not	Low

Appendix

C2: Touchpad evaluation (1st experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort
Mean	3,6667	4	3,9167	4,25	4,25	4
Standart Deviation	1,2309	0,8528	0,9962	0,866	1,1382	0,8528
Meaning	Comfortable	Smooth	Low	Accurate	Almost Acceptable	Comfortable

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Mean	3,4167	3,6667	3,9167	4,1667	4,4167
Standart Deviation	0,793	1,1547	1,0836	0,8348	0,9003
Meaning	Not high but not low	Low	Low	Low	Low

C3: Computer Mouse evaluation (1st experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort	Overall operation of input device
Mean	3,8333	4	3,8333	4,6667	4,6667	4	4,6667
Standart Deviation	1,1934	1,1282	1,3371	0,6513	0,4924	1,2792	0,4924
Meaning	Comfortable	Smooth	Low	Very Accurate	Acceptable	Comfortable	Very easy to use

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Mean	4,0833	3,5	4,3333	4,3333	4,3333
Standart Deviation	0,9962	1,3143	0,9847	1,2309	0,9847
Meaning	Low	Low	Low	Low	Low

Appendix

C4: Leap Motion (Touchless) evaluation (2nd experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort	Overall operation of input device
Average	2,7778	3,3333	2,1111	2,8889	3,2222	1,8889	2,5556
Standart Deviation	1,4814	1,1180	0,7817	1,1667	1,0929	0,7817	0,8819
Meaning	Not uncomfortable but not comfortable	Smooth	High	Not innacurate but not accurate	Not unacceptable but not acceptable	Uncomfortable	Not hard but not easy to use

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Average	4,1111	3,5556	2,3333	2,7778	3,7778
Standart Deviation	1,3642	1,3333	1,0000	1,3944	1,4814
Meaning	Low	Low	High	Not high but not low	Low

C5: Leap Motion (Grabbing) evaluation (2nd experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort	Overall operation of input device
Mean	3,6667	3,3333	2,7778	3,0000	3,7778	2,6667	3,3333
Standart Deviation	1,5000	1,4142	1,2019	1,1180	1,3944	1,3229	1,5811
Meaning	Comfortable	Not rough but not smooth	Not high but not low	Not innacurate but not accurate	Almost acceptable	Not uncomfortable but not comfortable	Not hard but not easy to use

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Mean	4,0000	3,8889	3,3333	3,6667	4,3333
Standart Deviation	1,5811	1,7638	1,6583	1,2247	1,0000
Meaning	Low	Low	Not high but not low	Low	Low

C6: Computer Mouse evaluation (2nd experiment)

	Force required for actuation	Smoothness during operation	Effort required for operation	Accuracy	Operation Speed	General comfort	Overall operation of input device
Mean	4,7778	4,4444	4,3333	4,6667	4,4444	4,5556	4,8889
Standart Deviation	0,4410	0,7265	0,7071	0,5000	1,0138	0,5270	0,3333
Meaning	Very Comfortable	Smooth	Low	Very accurate	Acceptable	Very Comfortable	Very easy to use

	Finger fatigue	Wrist fatigue	Arm fatigue	Shoulder fatigue	Neck fatigue
Mean	4,8889	4,6667	4,2222	4,7778	5,0000
Standart Deviation	0,3333	0,5000	1,0929	0,4410	0,0000
Meaning	None	None	Low	None	None

2

Table of Contents