FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Shamanic Interface for computers and gaming platforms

Filipe Miguel Alves Bandeira Pinto de Carvalho

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Leonel Caseiro Morgado (*PhD, Habil.*)

Second Supervisor: António Fernando Vasconcelos Cunha Castro Coelho (*PhD*)

January 31, 2014

Shamanic Interface for computers and gaming platforms

Filipe Miguel Alves Bandeira Pinto de Carvalho

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: José Manuel de Magalhães Cruz (PhD)

External Examiner: Hugo Alexandre Paredes Guedes da Silva (PhD)

Supervisor: Leonel Caseiro Morgado (PhD, Habil.)

Second Supervisor: António Fernando Vasconcelos Cunha Castro Coelho (PhD)

January 31, 2014

Abstract

In pursuit of overcoming the limitations of controlling digital platforms by imitation, the idea of integrating a person's cultural background and gesture recognition emerged. This would allow the creation of gestural abstractions to symbolize commands that cannot be completed, for any kind of impossibility, including physical impossibility, such as the case of disabled people or space constraints, as not every activity is adequate for closed environments.

Therefore, we propose a new approach on human-computer interaction: the shamanic interface. This proposal consists in using one's cultural background united with gesture recognition to develop a proof of concept to support future works in the area, using real-time gesture recognition and cultural richness to overcome some limitations associated to human-computer interaction by command imitation. This proposal aims at describing the challenges in the fields of natural interaction, gesture recognition and augmented reality for this work, alongside the cultural component.

After the analysis of the related work, there were no references to gestural recognition systems which included cultural background to overcome the limitations of command imitation, although some systems included the cultural component. Besides, it was possible to conclude that Microsoft Kinect is, at the moment, an adequate capture device for implementation of a natural gesture recognition system, because it only requires a camera to track the movements of the user. Microsoft Kinect tracking is also considered imperfect to track dynamic body movements, so the improvement of the skeletal tracking poses a challenge in the future. It is therefore expected that Microsoft Kinect 2.0 improves this tracking and detection to facilitate further developments on this area.

Through the implementation of the proof of concept cultural gestural recognition system, it was possible to conclude that gesture recognition systems have much room to evolve in the next years. The inclusion of the cultural background of the user provided improvements on the interaction, but the testing phase still needs some more time and tests with groups of users to reinforce its importance.

As a proof of concept, it is important to consider the wide diversity of paths to explore in the future in this area, because there is much attached to the shamanic interface left for future research.

Keywords: Natural Interaction, Culture, Gestural Recognition, Human-Computer Interaction.

Resumo

Com o objetivo de superar as limitações do controlo de plataformas digitais por imitação, surge a ideia de integrar a formação cultural de um indivíduo com o reconhecimento de gestos. Neste sentido é permitida a criação de abstrações gestuais de forma a simbolizar comandos que não podem ser efetuados, por qualquer tipo de impossibilidade, sendo que tal inclui a impossibilidade física, como no caso das pessoas com deficiência, ou restrições de espaço, pois nem todos os gestos são adequados para ambientes fechados.

Assim, foi proposta uma nova abordagem sobre a interação humano-computador: a interface xamânica. Esta proposta consiste em considerar a formação cultural de um indivíduo, conjuntamente com um sistema de reconhecimento gestual, para desenvolver uma solução baseada em reconhecimento de gestos em tempo real e a riqueza cultural de cada um para superar algumas limitações associadas à interação humano-computador por imitação de comandos. Esta proposta tem como objetivo descrever os desafios existentes nas áreas de interação natural, reconhecimento de gestos e realidade aumentada, sendo estas abordadas recorrendo à transversalidade inerente à componente cultural.

Após uma análise do estado da arte, concluiu-se não haver referência a nenhum sistema que inclua o background cultural de um indivíduo para superar as limitações da imitação de comandos. Além disso, foi possível afirmar que o Microsoft Kinect é, neste momento, um dispositivo de captura adequado para a implementação deste sistema de reconhecimento de gestos naturais, pois requer apenas uma câmara para acompanhar os movimentos do utilizador, sendo portanto simples para o utilizador comum, indo de acordo com o propósito de interação natural. O rastreamento através do Kinect é considerado imperfeito para acompanhar gestos dinâmicos, por isso surge também o desafio de melhorar o rastreamento do esqueleto para que o projeto possa ser considerado bem sucedido.

Através da implementação da prova de conceito, o sistema de reconhecimento gestual que tem em conta o background cultural do indivíduo, foi possivel concluir que os sistemas de reconhecimento gestual têm muita margem de evolução nos próximos anos. A inclusão da camada cultural permitiu assim melhorias a nível da interação, apesar de se tornar necessária uma fase de testes com utilizadores reais, de forma a salientar a sua importância.

Como uma prova de conceito, foi também importante encontrar os diversos caminhos para explorar futuramente na área, dado que há muito em falta no conceito da Interface Xamânica para investigação.

Palavras-chave: Interfaces Naturais, Cultura, Reconhecimento Gestual, Interação Humano-Computador.

Acknowledgements

I would like to thank to all the people that contributed somehow for this thesis and specially to my supervisors António Coelho and Leonel Morgado for their help, advices and for challenging me during this dissertation.

Filipe Miguel Alves Bandeira Pinto de Carvalho

"You don't understand anything until you learn it more than one way."

Marvin Minsky

Contents

List of Figures

List of Tables

Abbreviations

AR Augmented Reality

CLI Command Line Interface

DTW Dynamic Time Warping

GPS Global Positioning System

FAAST Flexible Action and Articulated Skeleton Toolkit

GUI Graphical User Interface

HCI Human-Computer Interaction

HMM Hidden Markov Model

ICT Information and Communications Technology

NI Natural Interaction

NUI Natural User Interface

RGB Red Green Blue

SDK Software Development Kit

UI User Interface

VR Virtual Reality

WIMP Windows, Icons, Menus, Pointers

WPF Windows Presentation Foundation

Chapter 1

Introduction

1.1 Context and Motivation

Through the years, Human-Computer Interaction (HCI) has been evolving. The Windows, Icons, Menus, Pointers (WIMP) paradigm reached masses and is probably the most common Graphical User Interface (GUI) paradigm. Recently, Natural User Interface (NUI) arose and the concept of Natural Interaction (NI) is getting more and more trendy, as there is much to develop, study and improve in the area. New approaches of interaction are arising to turn it into a continuously more natural interaction.

The world is currently under a phenomenon called globalization, making it easier to interact with people from different cultures or background. Even before, the idea of including culture to improve a system's interaction has been discussed: coherent behaviour of an application according to the user's cultural background can have a great impact on improving this interaction [RL11].

With all the technological advances, new devices are being created and, consequently, the popularity of natural interaction or, more specifically, gesture-based interaction is increasing. The launch of several devices, such as the Leap Motion and Myo support this idea.

The Leap Motion[1] is on the market since the end of July 2013. The developers claim that the device is 200 times more accurate than the other existent motion devices. The interaction is done by gestures in the upper field of the device, as observable in Figure 1.1. Leap Motion also allows the tracking of individual finger movements up to 1/100 of millimeter [LM213, RBN10].

In the early 2014, a special bracelet named Myo[3] will be launched. Developers claim that this bracelet will "use the electrical activity in your muscles" to control different kinds of electronic devices, like computers and mobile phones. The purported usage is expressed in Figure 1.2 [TL213].

[1]More information on: https://www.leapmotion.com/
[2]Source: https://www.leapmotion.com/product
[3]More information on: https://www.thalmic.com/myo/
[4]Source: https://www.thalmic.com/myo/

Figure 1.1: Leap Motion purported usage.[2]

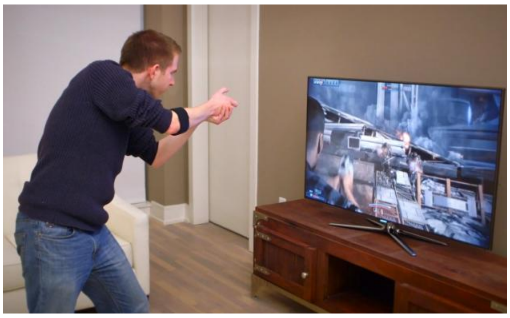

Figure 1.2: Purported use of Myo in a shooter game.[4]

Devices such as these will eventually revolutionize the market and the interaction between human and machines.

Most of this interaction is done by mimicry which has various limitations, from learning (see Figure 1.3 to observe the purported usage of Microsoft Kinect), as not all the commands are simple to imitate, to the use by the disabled population (see Figure 1.4), because the handicapped have several difficulties that harden the use of natural interfaces. These limitations can be hard to overcome and physical disabled persons tend to be dissociated from natural interaction, but some alternatives can be provided to allow the interaction to be more inclusive.

Figure 1.3: Learning is an issue on NI.[5]

Figure 1.4: Disabled people must be a concern on the development of a NI application.

Then, it arises the concept of the shamanic interface.

[5] Source: http://www.kotaku.com.au/2010/11/review-kinect-sports/

"It's called the shamanic interface because it was designed to be comprehensible to all people on earth, regardless of technological level or cultural background" [Sua10].

The idea of the shamanic interface was proposed by Daniel Suarez, a computer science professional and novelist, in his novels *Daemon* and *Freedom*™. This idea uses the vast group of somatic gestures together with shamanism, showing that even in primitive cultures there is something beyond the body, which is culture. Therefore the idea proposed in this dissertation is an interface usable by everyone regardless of each person's cultural background, uniting culture and gesture recognition. This means that the cultural background of the user is taken into account, on an analysis to explore the reduction of problems related with usability, learning curve and accessibility [Mor13]. Therefore, the scope of this dissertation includes gesture recognition, HCI and natural interaction.

To support this idea, this thesis included the development of a solution: a gesture recognition system using Microsoft Kinect[6]. In this approach, not only gesture recognition was important, but also the use of convenient gestures for meaningful actions. This required not only the study of gesture recognition, but an analysis of meaningful gestures for actions with logical feedback to use in the prototype application.

The main areas involved in this work are expressed in Figure 1.5. It is also worth of note, the extensive role of applications of the developed prototype: Augmented Reality (AR), virtual reality environments and remote control of applications using gestures.

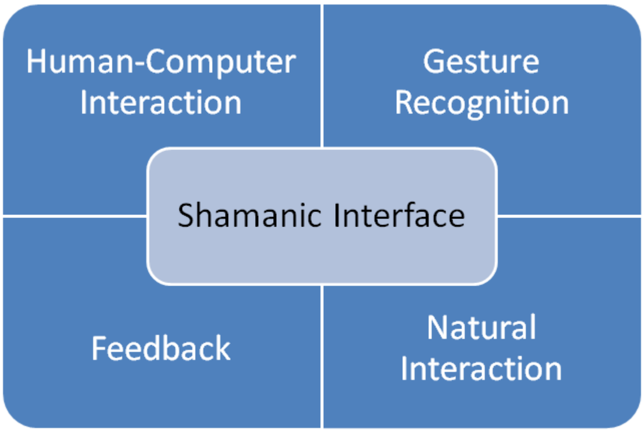

Figure 1.5: Context.

The main motivation of this dissertation consists in exploring how an individual's cultural background can influence the abstractions to command the machine when a certain instruction is not imitable or when it is hard to imitate. Giving the situation of a gestural-commanded system, an abstraction would be necessary to allow the user to control the machine. In consequence, many other motivations arise, such as creating meaningful abstractions for the controls, establishing

[6]More information on: https://www.microsoft.com/enus/kinectforwindows/

good usability patterns and making the system available for anyone, even for physically disabled users.

1.2 Research Problem: Statement and Delimitation

Natural interaction through gestures by mimicry can be greatly improved by analysing the actual gestures and searching for more meaningful solutions. However, such approach is not easy to achieve when commands cannot be imitated, which means that there is not an equivalent somatic movement, thus making the search for significant gestural abstractions harder. Consequently, gestural abstractions tend to be unnatural, going against the idea of natural interaction and, therefore, not taking advantage of it to attract new public.

It is important to balance functionalities and usability, as in any commercial product, to assure a good user experience and a good alternative to the usual human-computer interaction, for example, as in WIMP paradigm [GP12].

As explored in Section 1.1, HCI using gestures is a trending topic nowadays and has been a research focus during the last years. New techniques are appearing, such as recognition through wireless signals [PGGP13], new interaction devices are being launched (see Figures 1.1 and 1.2), but there is currently no knowledge of any gesture recognition application using the user's cultural background to overcome some of the inherent limitations of HCI by gesture imitation.

As the developed application is a proof of concept, another point is proving its relevance to allow and prepare a strong basis for future works on the area.

Nowadays, in HCI systems controlled by mimicry there are not meaningful alternatives for impossible to imitate commands. The problem emerges when some commands cannot be mimicryed.

Each platform creates its own conventions to symbolize the instructions for the user to control the system. In Figure 1.6, it is expressed the gesture used to interrupt Microsoft Kinect. In fact, this is a convention defined during the development that does not mean anything to the user: it requires the learning of a new gesture for the user to master the device. It is important to find meaningful gestures to be abstractions for impossible or difficult to imitate commands, which are the ones without a somatic equivalent. Physically disabled people were also a target of the application developed during this dissertation, as it is important that interfaces are inclusive and prepared to deal with people with movement limitations. This is one of the limitations of the majority of NUI and this thesis intends to work on that matter too.

In sum:

- In most of the situations, there are no alternatives to impossible or hard to imitate commands;
- Generally, when an abstraction exists, it is not meaningful for the user;
- The majority of NUI devices is not prepared for people with physical disabilities [KBR+07, GMK+13];

[7] Source: https://www.microsoft.com/enus/kinectforwindows/

Figure 1.6: Gesture to interrupt Kinect.[7]

- There is not a set of predefined gestures to constitute as general abstractions for HCI gestural-based devices.

Based on these ideas, there are some research questions that arise:

- How are gestures mapped into actions?
- What perspectives are relevant in gesture analysis?
- Are there gestures that can constitute abstractions for human-computer interaction based on gestures when commands are hard or impossible to imitate?
- Are they meaningful?
- Is it possible to integrate culture to consider this interaction meaningful?

These are the main questions in which we intended to contribute during the development of this dissertation and are explained in Chapter 3. The mapping of gestures into actions was done through the use of keyboard strokes. The analysis of gestures took into account different perspectives such as rhythm and amplitude and therefore some gestures were selected to illustrate the concept of the shamanic interface.

The work in this dissertation followed the Design Science methodology, which is explained in Section 1.3.

1.3 Methodology

The Design Science methodology includes three simultaneous cycles: Relevance Cycle, Design Cycle and Rigor Cycle, as we can observe on Figure 1.7.

The Relevance Cycle includes the analysis of the relevance of the problem, includes the analysis of the requirements of the implementation and all the testing of technologies to use.

The requirements of the implementation would be the first step if the work was purely practical. Using Design Science approach, work is continuous and also simultaneous, therefore, the three cycles occur intercalated. Using the data gathered by analysing other gesture recognition

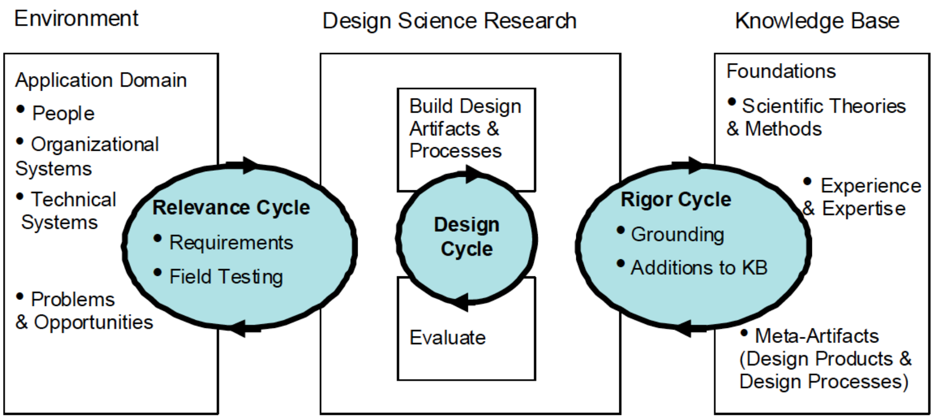

Figure 1.7: Design Science [Hev07].

systems, it was possible to detail the requirements for the proof of concept, except in the culture integration aspect. On that aspect, the study on culture based systems, helped to understand important aspects on a cultural background and how to connect them with a gesture recognition system. The idea of creating a layer between the capture and the recognition allowed to fulfil this purpose and maintaining intact the main concepts of a recognition system.

Later, it was time to search for technologies that could help to achieve the purported usage. More important than the technologies were the devices. Therefore, an analysis of Nintendo Wii Remote and Microsoft Kinect were essential. The use of recent devices, such as Myo and Leap Motion, was discarded because of not being available while Microsoft Kinect was available. It is also worth note that the development phase started in September, so Leap Motion was only on the market for one month, not providing a real alternative to the named devices. This search and analysis is more detailed in the Chapter 2.

The Design Cycle includes all the implementation process and evaluation of the developed product. The implementation is a prototype of the cultural-based gesture recognition system. It uses Microsoft Kinect for tracking user's movements. More details can be found in Chapter 3. The continuous evaluation of the developed system was done by several experiences during the development of the proof of concept system to understand the best way to express the concept of the integration of cultural background on a gesture recognition system. This part of the project is explained in detail on Chapter 4.

As for the Rigor Cycle, there has been work on analysing the relevant state of the art to add to the knowledge base, allowing the work to improve some points on what has been done in the area. The knowledge base includes all the information gathered during the development of this dissertation, culminated in this writing. The first step included the full understanding of the problem involved, which included several phases: problem definition, objectives of the work to realize and the involved areas.

It was important the study of the evolution on the fields of gesture recognition, human-computer interaction and natural interfaces, so the study of the state of the art was the next step. The literature in the area was vast and a selection of the most suitable articles was chosen to express the more relevant points on the development of this work.

This study of the literature would not be complete without analysing gesture recognition using Kinect and several important projects in the area, such as FAAST and Online Gym, referred in Chapter 2. Again it was impossible to study all the projects in the areas, but a selection was done to cover different projects, each one important somehow.

Design Science methodology is also based on guidelines, which can be seen in Table 1.1.

Table 1.1: Design Science Research Guidelines. Adapted from [HMPR04].

Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

These instructions helped seek guidance during the development of this dissertation. They also function as a complement to the Design Science cycles, exploring deeply the results of the development process during these cycles.

The evaluation of the system was done according to the Design Science methodology, which we can observe in Table 1.2. Given the time constraints, not all methods were possible to apply. Still, some were applied and are very important metrics to measure the quality of the work [HMPR04]. The evaluation of this work is explained with more detail in Chapter 4.

1.4 Expected Contributions and Main Goals

Besides the theoretical study on the connection of culture with gestural interaction, the objective of this project is to implement a gesture recognition system prototype, including a set of trained gestures, with different intents, to serve as a proof of concept to evaluate the impact of an individual's cultural background in the creating of meaningful gestural abstractions to substitute impossible or hard to imitate commands in HCI. The use of specific culture related gestures would require a deeper study that would be utterly constrained by the time restrictions affecting this dissertation, therefore, the use of these defined gestures substitute them for the basic solution developed. The used gestures are not of high complexity but symbolize several different gestures used by people with diverse backgrounds.

As the project intends to support future works in this area, it is explored the connection between an individual cultural background and the creation of gestural abstractions as alternatives to commands that cannot be imitated. Besides, a gesture recognition system was also implemented to illustrate the concept.

Table 1.2: Design Science Evaluation Methods. Adapted from [HMPR04].

1. Observational	**Case Study**: Study artifact in depth in business environment **Field Study**: Monitor use of artifact in multiple projects
2. Analytical	**Static Analysis**: Examine structure of artifact for static qualities (e.g., complexity) **Architecture Analysis**: Study fit of artifact into technical IS architecture **Optimization**: Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior **Dynamic Analysis**: Study artifact in use for dynamic qualities (e.g., performance)
3. Experimental	**Controlled Experiment**: Study artifact in controlled environment for qualities (e.g., usability) **Simulation**: Execute artifact with artificial data
4. Testing	**Functional (Black Box) Testing**: Execute artifact interfaces to discover failures and identify defects **Structural (White Box) Testing**: Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation
5. Descriptive	**Informed Argument**: Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility **Scenarios**: Construct detailed scenarios around the artifact to demonstrate its utility

1.5 Dissertation Structure

Besides Chapter 1, the Introduction, this document is composed by four more chapters.

Chapter 2 describes the concept of NI, an introduction to the process of Gesture Recognition and aims to present the state of the art in the natural interaction and gesture recognition fields. It also presents some applications already developed in the area.

Chapter 3 contains the description of the proposed solution as well as the tools this solution is based on.

Chapter 4 includes the analysis of the obtained results from the study of the topic and the developed system, according to the Design Science methodology.

Chapter 5 presents the conclusions of this work and also some of the next steps available for research to continue in this area.

Chapter 2

Literature Review

This chapter includes the background and related work of this thesis, beginning with some definitions used in this dissertation and then the study done about natural interaction and gesture recognition.

At first, there is the need to define gesture, commonly used in this work:

> *"A gesture is a form of non-verbal communication or non-vocal communication in which visible bodily actions communicate particular messages, either in place of, or in conjunction with, speech. Gestures include movement of the hands, face, or other parts of the body. Gestures differ from physical non-verbal communication that does not communicate specific messages, such as purely expressive displays, proxemics, or displays of joint attention"* [Ken04].

There is a problem in the relationship between humans and technology-enhanced spaces. The use of adequate interfaces is therefore very important as this interaction is, in most cases, extremely simple. Nowadays, interfaces tend to be too complex even if their objective is very simple. Design practices are still very attached to the WIMP paradigm.

> *"The use of controllers like keyboards, mouse and remote controls to manage an application is no longer interesting and therefore, are getting obsolete"* [RT13].

WIMP-based technologies are tendentiously difficult to use and require training. The closer interfaces are to the way people naturally interact in everyday life, the less training and time is spent to correctly use the system [YD10].

The focus on simplicity also allows a greater target public on an application because,

> *"The higher is the level of abstraction of the interface, the higher is the cognitive effort required for more interaction"* [Val08].

> *"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them"* [HBC+96].

According to Valli, people naturally use gestures to communicate and use their knowledge of the environment to explore more and more of it [Val08]. This is the definition of natural interaction, that, as a secondary objective, is aimed by this project: improve usability of human-computer interaction using gestures.

The naturalism associated to gesture communication must be present in human-computer interaction applications: the system shall recognize systems the humans are used to do [KK02]. Adding the idea of cultural background, gesture become even more natural to the user, as they reflect knowledge associated with himself. This reasoning also works for interaction: more familiar interfaces will have a smaller learning curve than different approaches, as Valli states:

> *"Designing things that people can learn to use easily is good, but it's even better to design things that people find themselves using without knowing how it happened"* [Val08].

2.1 Interfaces and Interaction

Recently, various systems emerged that ease gesture and motion interaction. As time goes by these systems tend to become mainstream and it urges the need of an implicit, activity-driven interaction system [LARC10].

Interfaces have been evolving over the years: the first user applications were command-line based, usually known as Command Line Interface (CLI). Static, directed and abstract interfaces that only use text to communicate with the user. An example can be seen in Figure 2.1, in a command-line interface application of the board game Camelot.

Some years later, the first GUI appeared, using graphics to provide a more interactive and responsive User Interface (UI). An example can be seen in Figure 2.2, which shows a GUI. The problem arises as sometimes GUIs are not very intuitive and HCI had still room to improve. There are studies about predicting the next move of the user, using the eye movement as an example of the importance of improving interfaces and the potential present in the area [BFR10].

Despite the evolution, GUIs were still not very direct, what leaded to the appearance of NUI [FTP12]. Touch interfaces, as in Figure 2.3 and gesture recognition interfaces belong to NUI.

Natural user interfaces are the ones related to natural interaction, in which the user learns by himself. As Valli states, the use of gesture to establish communication is natural [Val08], therefore platforms which are based on gesture recognition are associated to natural interaction.

NUI tend to be direct, intuitive and based on context: the context tends to direct the user to know how to use the platform [NUI13]. These interfaces also intend to be a valid alternative to all the WIMP-based interfaces, which are generally considered to be hard to use and require previous training [YD10, FTP12]. This poses as an obstacle mainly to elderly individuals or people with difficulty to learn [RM10]. Valli states that:

> *"Simplicity leads to an easier and more sustainable relationship with media and technology."* [Val08]

This can help surpassing this issue by turning the interaction easier and more simple. This new kind of interaction is also an advantage as it motivates individuals to concentrate only on the task to perform and not in the interface itself [YD10]. The desire of creating the NUI "has existed for decades. Since the last world war, professional and academic groups have been formed to enhance interaction between "man and machine" [VRS11].

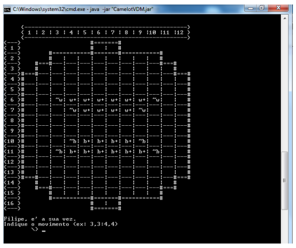

Figure 2.1: Camelot board game as a CLI application.

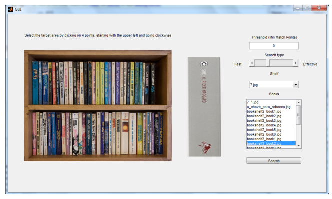

Figure 2.2: GUI of a book visual recognition application.

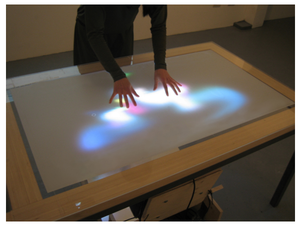

Figure 2.3: Natural interface using touch.[1]

It is also important to define the relevant concept of inclusive design. According to Reed and Monk, inclusive design must engage the widest population possible, not only future but also actual. One important point on achieving inclusive design is addressing people with disabilities, as these tend to be excluded by technological evolution [RM10].

User Interaction is exploring new paths and, in many applications, moving away from WIMP to more physical and tangible interaction ways [SPHB08].

As Champy states,

> *"The nature of the interaction between the controller and the UI historically has levied unnatural constraints on the user experience"* [Cha07].

Natural Interaction provides the simplicity of interaction with machines without additional devices and is available for people with minimal or null technical knowledge [CRDR11]. The communication with the system is also enhanced as it is intuitive which diminishes the learning curve.

In Figure 2.4 there is the representation of an action triggered by the pressing of a button. On the other hand, on a gesture recognition system, an action can be triggered by a specific movement (see Figure 2.5, instead of a specific button, the trigger is a movement). Therefore, the mapping between action and feedback from the system is done similarly, but the performed actions are different.

[1] Source: http://www.skynetitsolutions.com/blog/natural-user-interfaces

Standard controllers weakly replace natural interactions. Therefore they all require a learning curve for the user to be used to it. A good improvement in this situation is reducing to the minimum this learning curve [Cha07]. In the prototype application, the idea of using meaningful gestures for certain actions is very important.

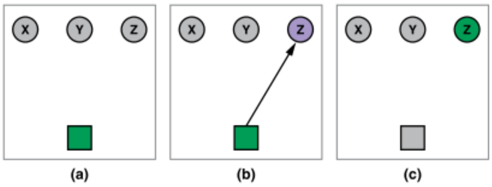

Figure 2.4: Pressing a button substitutes the natural physical interaction [Cha07].

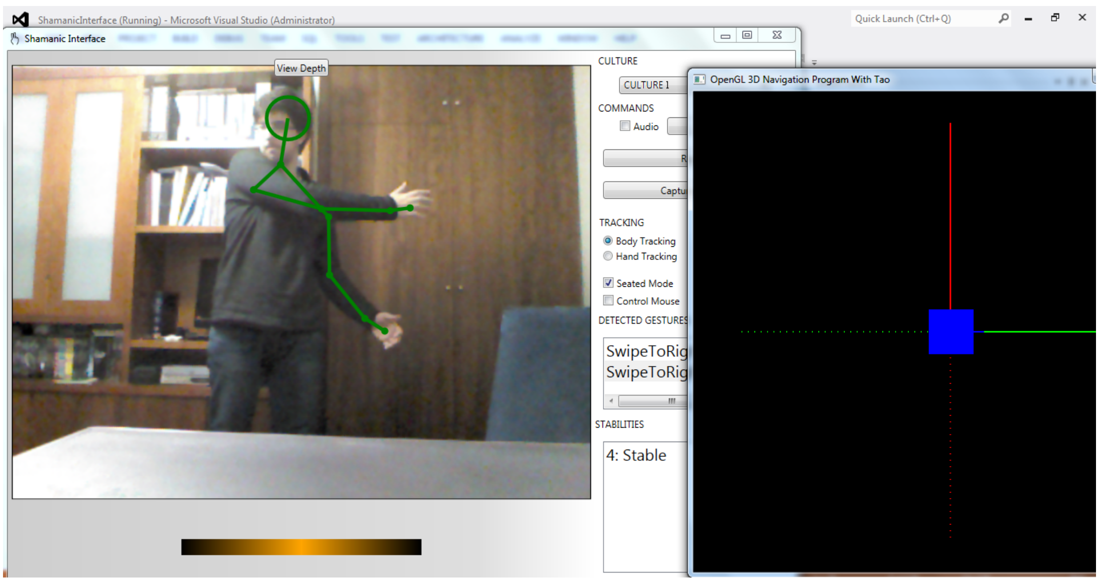

Figure 2.5: A movement used to substitute pressing a button.

Motion gaming is also a trendy topic and very relevant for this research as this idea can also be used for games and generic controls. As referred in [Cha07], the objective is that the prototype is intuitive and so it arises intuitive design, which intends to use gestures and movements a person uses on its daily routine and bring them to the interaction with the system.

The use of RGBD cameras, such as Microsoft Kinect turned the image processing job in determining relevant features for gesture classification easier [IKK12]. The complexity of tasks is a very important point on learning. The moment learning is needed is when pure trial and error is too exhaustive [CFS+10].

2.2 Culture-based Interaction

"Culture influences the interaction of the user with the computer because of the movement of the user in a cultural surrounding" [Hei06].

This statement shows us the importance of the culture background of each person to provide the best interaction possible in a system. Therefore, as the system is intended to be available for the widest population possible, the differences on how people from different cultures interact is very relevant for this area.

Using Kinect, Microsoft showed the way to controller-free user interaction [KED+12]. By controller-free, it is considered the use of devices not coupled in the body of the user or remote controls like the ones in gaming platforms.

A different approach according to cultural beliefs requires the system to be able to recognize culturally-accepted gestures. Only recently there has been investigation about the integration of culture into the behaviour model of virtual characters. Speed and spatial extent can also be indicators of an user's culture and that's considered an important detail to build a stronger application [KED+12].

Works in the area tend to use virtual characters to represent the exact movement of the user. An example is the Online Gym project referred in [CFM+13] that intends to create online gym classes using virtual worlds.

Recent studies evidence that users tend to enjoy the interaction with the Kinect which takes to a growing interest in using the system to perform the interaction. As the age range is wide, it diminishes the possibility of arouse interest in younger users, which leads to an application for a vast target public [KED+12].

Rehm, Bee and André state that:

> *"Our cultural backgrounds largely depend how we interpret interactions with others (...) Culture is pervasive in our interactions (...)"* [RBA08].

On Figure 2.6 the differences in a usual waiting posture between a german and a japanese can be observed. These postures tend to have a cultural heritage and are therefore considered part of the cultural background of the users.

Figure 2.6: Differences in postures from people with different cultural backgrounds [RBA08].

2.3 Augmented Reality

AR is one of the several targets of this application. The use of gesture recognition systems in AR applications already exists, which support the interest in using the system for that purpose.

Therefore, it is important to analyse briefly the literature on this theme.

According to Duh and Billinghurst:

> *"Augmented Reality is a technology which allows computer generated virtual imagery to exactly overlay physical objects in real time"* [DB08].

Augmented Reality's (see Figure 2.7) objective is to simplify the user's life by combining virtual and real information on the point-of-view of the user [CFA+10].

Figure 2.7: Augmented Reality between virtual and real worlds [CFA+10].

The first system in AR was created by Sutherland using an optical see-through head-mounted display.

Tracking and interaction are the most trendy topics on AR, according to Duh and Billinghurst [DB08]. This analysis is based on the percentage of papers published and paper citations.

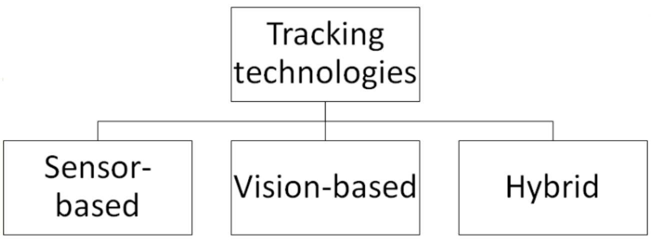

Figure 2.8: Differentiation on tracking techniques. Adapted from [DB08].

The diagram presented in Figure 2.8 shows the different modalities on AR tracking. Sensor-based tracking is more detailed in some entries of the Table 2.1.

Table 2.1: User tracking device types. Adapted from [KP10].

Type	**Example Device**
Mechanical, ultrassonic and magnetic	Head-mounted display
Global positioning systems	GPS
Radio	RFID
Inertial	Accelerometer
Optical	Camera
Hybrid	Gyroscope

On the other hand, vision-based tracking includes tracking using image or video treatment and also marker-based tracking. Markers are unique identifiers can be barcodes, radio-frequency

(RF) tags, tags or infrared IDs. These are tangible and physically manipulable. Different technologies have different pros and cons: infrared need batteries and RF tags are not printable for example [RA00]. The different usable tags are expressed in Table 2.2.

Table 2.2: Tag comparison. Adapted from [RA00].

Type	**Visual Tags**	**RF Tags**	**IR Tags**
Printable	Yes	No	No
Line-of-sight	Required	Not Required	Required
Battery	No	No	Required
Rewritable	No	Yes/No	Yes/No

Markerless and non-wearable devices are less intrusive solutions which are more convenient for real-world deployments.

The Figure 2.9 represents the tracking process in an AR system. At first, there is the tracking part and then the reconstruction part of the process, combining both the real world and virtual features.

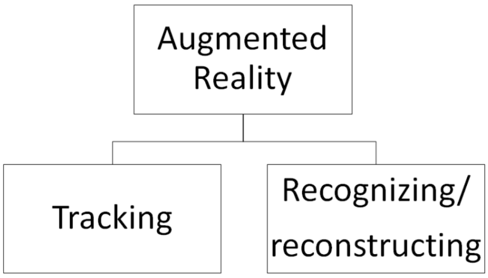

Figure 2.9: Augmented Reality tracking process description. Adapted from [CFA+10].

Such as in gesture recognition system, also in AR the fast processing is very important to allow immersion of the user and a more reliable response system [DB08].

Current AR systems rely heavily on complex wearable devices, such as head-mounted displays, as referred in Table 2.1. These devices also tend to be fragile and heavy and therefore not suitable for frequent use.

Data gloves are also not appropriate for everyday interaction, because their use is not comfortable or even natural, so they are only adequate for casual situations. They restrict the use of hands in real world activities and limit one's movements. Nevertheless, hand movement may be tracked visually without additional devices. For gesture recognition, the use of cameras is advantageous, because it does not restrict hand or body movements and allow freedom [KP10].

2.4 Gesture Recognition

Figure 2.10 refers the differences in gesture types used in this thesis. At first, static gestures are commonly referred as postures, as they describre specific relations between each one of the tracked joints. Gestures that include linear movement are the ones in which one or more joints are moved in one direction with associated speed. Complex gestures depend on the tracking of one or more joints that move in non-linear directions over a certain amount of time [KED+12].

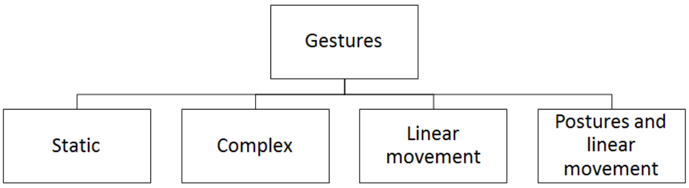

Figure 2.10: Gesture differentiation. Adapted from [KED+12].

In Table 2.3, the difference between static and dynamic gestures can be observed through examples. It is important to fully understand the notion and difference on gestures. Another common denomination for static gestures is postures, as they do not rely on movement but poses. On the other hand, dynamic gestures rely on the movement realized.

Table 2.3: Static and their dynamic gesture counterparts. Adapted from [Cha07].

Static gesture	**Dynamic gesture**
Hands together	Clap hands
Raise one arm	Wave arm
Arms to the side	Pretend to fly
One-leg stand	Walk in pace

According to Yin and Davis, hand movements can be divided in [YD10]:

- Manipulative gestures - used to interact with objects
- Communicative gestures - used between people

On the other hand, according to Krahnstoever and Kettebekov, gestures can be part of another classification [KK02]:

- Deictic gestures - strong dependency on location and the orientation of the hand
- Symbolic gestures - symbolic, predefined gestures, such as the ones in sign language or cultural gestures

Symbolic gestures are the ones targeted on this work, because of their symbolic meaning. Meanwhile, deictic gestures are standalone gestures that strongly depend on the context.

To correctly analyse gesture recognition, isolated gesture recognition and continuous gesture recognition must be separated. The second part depends greatly on achieving the first with accuracy [YD10].

Another important issue is related to the real-time interaction. The recognition must be fast enough to allow real-time interaction because a gesture interaction system demands it in order to achieve a good interaction and pose as a real alternative to other HCI systems.

There is a lot of work reported for gesture recognition. Sometimes the tracking refers to the user's full body [MMMM13, GLNM12] but, regarding gesture recognition the most important component is hand tracking [Li12a, Li12b]. Despite having analysed hand tracking, in this dissertation the focus was full-body motion.

Gesture Classification Gesture classification is a studied topic, but it is not clear which is the best classificator. The most used classificator are Hidden Markov models, but there are other approaches, such as using Dynamic time warping or Ant recognition algorithms [ABD11].

Algorithms Hidden Markov Model (HMM) is a very useful algorithm for isolate gesture detection. It is used as a classification machinery using a variety defined as Bakis model. This model allows transitions between several states, compensating different gesture speeds. This task becomes therefore important as different people tend to realize gestures at different speeds [YD10]. For each gesture, there is one HMM. The probability of a certain gesture sequence to be confirmed as a recognized gesture is based on a model that gives the highest log-likelihood [YD10].

Feature vectors are also widely used in the area. The tracker supplies data describing joints localization in x,y and z coordinates and the orientation of each joint. According to Yin and Davis, this method has a great application on hand gesture recognition and so the improvement of using it for body movement recognition would be done similarly [YD10].

Continuous gesture detection requires segmentation. Segmentation is widely used to allow the recognition of a continuous gesture sequence, as it permits the detection of the beginning and the end of the gesture, in order to know which segment of the movement to classify [YD10].

Gesture Training As for the comparison of the detected gestures, a gesture recognition system must have a set of trained gesture to realize the comparisons. These gestures are generally stored in the form of relevant data (orientation, hand position in the 3 axis and velocity in the plane) and can be stored using various notations, such as XML or JSON.

Hand Detection Skin color detection is a very common method used for hand localization. It filters the hand through the color of the skin of the user, but it has issues related to the tracking of other parts of the body, because the skin has the same color over all the body [CRDR11].

Depth thresholds use the different euclidean distances between the user and the camera and the background and the camera to filter the hand [Li12a, Li12b].

The combination of these two methods can pose as a more accurate solution to prevent errors and filter the points of interest of the image. As Red Green Blue (RGB) images are not suited for an accurate feature extraction, the conversion for binary or intensity images gives better results.

Hand Orientation It is relevant to recognize the hand position at each moment, so feature vectors are frequently used to store that information. As Chaudhary et al. state, the tracker supplies data describing the hand coordinates and orientation and the vector saves the velocity of the hand in a plane, such as xy and the exact position the other axis, such as z. Keeping this information at each moment, allows a later comparison between the recognized gestures and the trained gestures in the application [CRDR11].

Another alternative to allow the recognition, despite scale and rotation is the use of homographs, but it would require more complicated operations in real-time which would lead to performance problems.

Gesture Comparison The use of machine learning algorithms is very important for gesture classification. As aforementioned, Hidden Markov Models are used as classification machinery. They generate one model for each trained gesture and store each state of the movement to compare with the sets of trained data of the system. One important issue is to compensate different gesture speeds, because users do not perform the gesture with the same exact velocity. The probability of an observed sequence of gestures is then evaluated for all the trained models with a classification on the highest likelihood [YD10].

Other approaches are possible, such as the one used by simpler gesture detection systems, such as Kinect Toolbox, which defines the gesture through constraints. Therefore, if the movement is according to the limitations, the gesture is valid. This approach is detailed on Chapter 3.

2.5 User Movement Tracking

In gestural recognition systems, accuracy is essential in user movement tracking devices. This is utterly important, because this is a real-time system.

Freedom of movements is also of top importance when dealing with HCI applications. Successful HCI systems should mimic natural interaction humans and they are used in everyday communication and respond towards it [KK02].

This interaction may include distinct constraints, such as wires or other coupled devices which reduce and inhibit freedom of movement and orientation, reducing the will of the user to use them. Besides, additional devices become awkward to the user while gesturing and require users to learn how to deal with them, which includes a learning curve, that is not always short. For a NUI, it is important to consider that the shorter the learning curve, the better, because naturalism require easy-to-learn interfaces and simple ways of interaction.

Observing Table 2.1 again, it is presented a brief notion of the user tracking device types used nowadays for user movement tracking, mainly in AR systems.

- Head-mounted displays and other mechanical, ultrasonic or magnetic devices are mainly for indoor use, because of the equipment it requires the user to wear. Equipment like this is not proper for a common user to use, as it requires technical knowledge, not being natural as devices for natural interaction require. Besides, the use of this device implies the generation of virtual content. These devices are still widely used in Virtual Reality (VR) and AR fields.

- GPS are widely used for tracking in wide areas, but its precision - 10 to 15 meters - is not very useful for distinguishing user movements.

- Radio tracking requires previous preparation of the environment by placing devices to detect the radio waves. Once again, it requires technical knowledge and therefore is not directed to the common user. As a complement, wireless tracking is also used.

- Inertial sensors are widely used nowadays. Accelerometers are one of the most-used motion sensors and are present in a wide range of commercial products, such as smartphones, cameras, step counters, game controllers, capturing devices, etc. Among the capturing devices, its presence is utterly important in devices such as Nintendo Wii Remote[2] and smartphones. Inertial devices must be updated constantly on the position of the individual to minimise errors. One advantage of these sensors is that they do not need previous preparation nor technical knowledge to be used. Besides, its use can also be in monitorization, using body-worn accelerometers to track a person daily movements for example [LARC10].

- Optical tracking is usually based on cameras. This tracking is divided into two groups: marker-based and marker-less. The use of markers - fiducial markers or light emitting diodes - to register virtual objects is quite common because it eases the computation but it turns it less natural. Therefore, marker-less tracking is the objective, where the use of homographs to align frame to frame the rotation and translation images to realize its orientation and position.

- Hybrid tracking combines at least two of the other user tracking types and is nowadays one of the most promising solution to deal with the issues of indoor and outdoor tracking [KP10].

Just as technology is evolving, so are sensors. With the notorious increase of sensing devices integrated in commercial products and the evolution of sensing technologies, there is a facilitated path to a new generation of interactive application that also improve user experience [LARC10].

Acceleration can be from two types, according to [LARC10]: static acceleration which is the orientation with respect to the gravity and dynamic acceleration which relays on the change of speed. This division is present on the diagram, presented on Figure 2.11.

2.5.1 Motion Capture Devices

Recently, there has been an explosion of new low-cost body-based devices in the market. They include various applications, since medicine, sports, machine control and so on. These motion-based trackers, such as Microsoft Kinect, provide an opportunity for motivating physical activity [Cha07]. Games are an important part of motion-based controllers, because this is one of the most used ways of using motion-based interaction. They also promote physical and emotional well-being for elderlies and tend to motivate them to exercise.

As Champy states:

[2] More information on: https://www.nintendo.com/wii

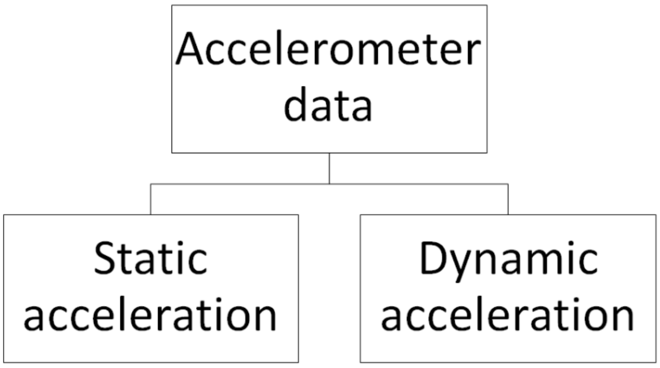

Figure 2.11: Accelerometer data. Adapted from [LARC10].

> *"As our population ages, our digital entertainment systems become more persuasive, we can expect interest in video games among older adults to increase"* [Cha07].

This gets utterly important as it allows older people to be a target public for motion-based applications. There is not many research on full-body motion control in videogames. Available work related has focused on compiling gesture recommendations and player instructions [Cha07].

The use of specific devices, such as Microsoft Kinect and Nintendo Wii Remote simplify the recognition process, not relying on the use of additional hardware.

The calibration of Kinect is realized with a predefined posture. When using Microsoft Kinect Software Development Kit (SDK), there is no need to calibrate the system: the user should be recognised instantly. On the other hand, the use of other frameworks such as OpenNI require previous calibration [ABD11].

The combination of different motion capture devices is also possible. As an advantage, the results of combining different devices can be better than using just one as the precision is improved.

One point against motion-capture device combination relays on the complexity inherent to the setup of all the system. The system becomes obviously more complicated and that is not acceptable to use for applications intended to be used by everyone. Technical knowledge is also a requirement and there are restrictions on the environment in which the user can use it [ABD11].

2.5.1.1 Nintendo Wii Remote

Nintendo Wii was launched on November 2006 and was the first that included physical interaction in their games (see Figure 2.12). This interaction is realized through the Wii Controller (Wii Remote or Wiimote), which can be observed on Figure 2.13, using accelerometer technology to detect the users' movements [SPHB08]. Wii is also a popular platform, as it has sold around 100 million consoles since launch [Sal13].

Wiimote is a good device for motion detection using accelerometer technology, because of its ease of use, price and ergonomic design. The controller is represented in Figure 2.13. Accelerometer technology relays on saving characteristic patterns of incoming signal data representing the

[3] Source: http://www.canada.com/story.html?id=5ff7f35b-e86b-4264-b3e6-19f6b5075928

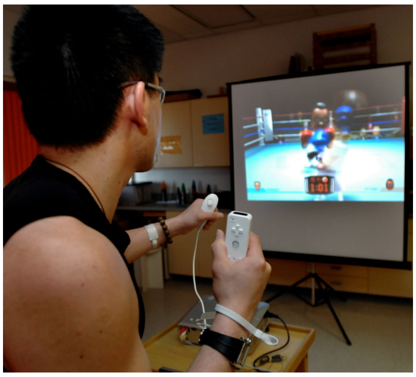

Figure 2.12: The use of Wii in a boxing game.[3]

controller in a tridimensional system of coordinates [SPHB08]. Wii Remote possesses an inertial device that can minimize faults by arms occlusion, unlike Microsoft Kinect. It communicates through a wireless Bluetooth connection, minimizing the uncomfort for the user. Therefore, Wiimote does not depend on wires, but restrains the user, as the device must be coupled with the user's arm, minimizing the naturalism of the movement. This device includes a three axis accelerometer, an infrared high resolution camera and transmits data using the Bluetooth technology. It is possible to develop application based on the use of the Wii Remote using libraries and frameworks available online [ABD11, FTP12, GP12, SPHB08]. There are some cases of using Nintendo Wiimote to control computer, such as [Wil09].

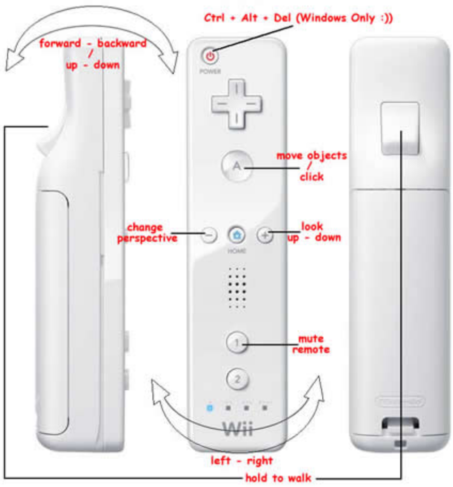

Figure 2.13: Wiimote controller.[4]

2.5.1.2 Microsoft Kinect

In November 2010, Microsoft Kinect was launched. Its real-time interaction with the users through the camera was the sign of the arising of new approaches on human-computer interaction [Li12a,

[4]Source: http://www.grantowngrammar.highland.sch.uk/Pupils/3E09-11/LucindaLewis/Inputdevices/Motion_sensing.html

Li12b, MS213]. Kinect sold around 24 million devices by February 2013 [Eps13], which makes it a very popular device for interacting with computers and gaming consoles, serving the purpose of this thesis, as shown in Figure 2.14.

Figure 2.14: Gaming using Kinect.[5]

Microsoft Kinect uses a RGB camera to track and capture RGB images and depth information between 0.8 to 3.5 meters of distance of the user [ABD11, SLC11]. Depending on the distance, the device is more or less precise, when the distance is 2m it is able to be precise until 3mm [VRS11]. Its architecture is detailed in Figure 2.15.

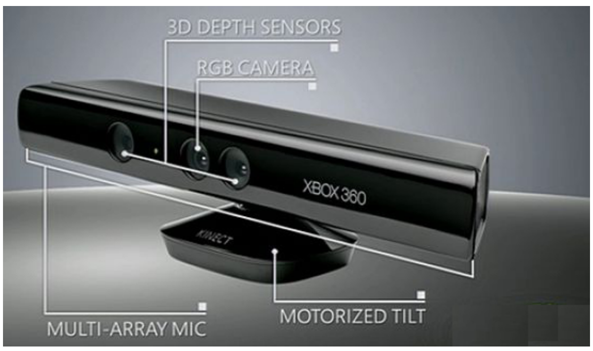

Figure 2.15: Architecture of the Kinect sensor.[6]

More specifically it returns an exact information about the depth and color of each point detected and the coordinates of each one of the detected points [Pau11]. Kinect processes the information without coupling any device to the user, which goes along with the purpose of NI [Val08]. Skeleton tracking using Kinect can originate some issues, as occlusion problems, which can be corrected with an improvement of the recognition or the use of additional hardware to complement Microsoft Kinect data [ABD11, BB11, MS213, GP12, Li12a, Li12b].

The technical specifications of Microsoft Kinect are detailed next: [MEVO12, ZZH13]

[5]Source: https://www.gamersmint.com/harry-potter-meets-kinect
[6]Source: http://praveenitech.wordpress.com/2012/01/04/35/

- Color VGA motion camera: 1280x960 pixel resolution
- Depth camera: 640x480 pixel resolution at 30 fps
- Array of four microphones
- Field of view:
- Horizontal field of view: 57 degrees
- Vertical field of view: 43 degrees
- Depth sensor range: 1.2m - 3.5m
- Skeletal Tracking System
- Face Tracking, track human faces in real time
- Accuracy: a few mm up to around 4cm at maximum sensor range

Despite the low cost of the Microsoft Kinect sensor, its results are very satisfactory. Nonetheless, Kinect may be considered by some as "of limited cutting edge scientific interest" [MMMM13], but with the various available tools to develop, it creates certain conditions adequate for scientific demonstrations. The low cost of the equipment is then considered a great advantage as many developers grow interest in the platform and developments on the area born and grow [MMMM13].

This year, 2014, along with the new console Xbox One[7], the new Kinect SDK will be launched with some improvements [Hed13]:

- Higher fidelity - The use of a higher definition color camera together a more accurate and precise system produce more loyal reproductions of the human body (see Figure 2.16).
- Expanded field of view - Minimizes the need to configure existent room for a better detection. Together with the higher fidelity eases and improves gesture recognition.
- Improved skeletal tracking - Increases the body points tracked and allows the participation of multiple users simultaneously.
- New active infrared (IR) - The new capabilities improve the resistance to light, improving the recognition capabilities independently of the environment.

It is important to analyse the retroaction of Kinect applications in the future to allow the system to take advantage of a new capture device and last longer.

[7]More information on: http://www.xbox.com/pt-PT/xboxone/meet-xbox-one
[8]Source: http://blogs.msdn.com/b/kinectforwindows/archive/2013/05/23/the-new-generation-kinect-for-windows-sensor-is-coming-next-year.aspx

Figure 2.16: The new Kinect purported recognition.[8]

2.5.1.3 Other Devices

Other devices were launched by the time, like Playstation Move[9], but their market expression and technological improvements were not that relevant.

A mention to the combination of different tracking devices is in both the works from [ABD11] and [DMKK12]. There is a trend on combining different devices that use different technologies to allow a more rigorous tracking. The use of a Kinect camera combined with two Wii Remote devices coupled to each one of the arms of the user to reduce faults by occlusion of the arms, which are a problem of using Microsoft Kinect's detection. The calibration is possible to unify, defining an unique calibration for both the devices [ABD11]. The problem arises in the use of a combination of several devices, including the Wii Remotes, which must be coupled to the arm of the user to be effective. This would cause the system to become more difficult to use and the user uncomfortable, which would go against natural interaction purpose [Val08].

It is worth mention that there are some relevant devices for HCI through gestures that will be launched in the next months, contributing to the growing interest in the area.

The Leap Motion[10] was released on the market in July 2013. This device is 200 times more accurate than the existent motion devices according to developers. The interaction is done by gestures in the upper field of the device (see Figure 2.17). Leap Motion also allows the tracking of individual finger movements up to 1/100 of milimeter [LM213, RBN10]. As the product is very recent, evaluations are still preliminary, but the recognition works only up to 60 cm distance. Therefore, this would pose as a problem to the recognition of somatic movements as the distance is very small not to restrict free movements of the user [Pin13].

In the early 2014, a special bracelet named Myo[11] is launched. This bracelet will "use the electrical activity in your muscles" to control different kinds of electronic devices, like computers and mobile phones [TL213] (see Figure 2.18).

[9] More information on: https://us.playstation.com/ps3/playstationmove/
[10] More information on: https://www.leapmotion.com
[11] More information on: https://www.thalmic.com/myo/
[13] Source: https://www.leapmotion.com/product
[13] Source: https://www.thalmic.com/myo/

Figure 2.17: Leap Motion purported usage.[12]

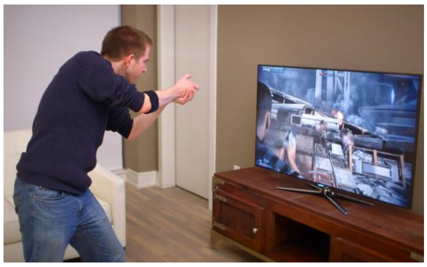

Figure 2.18: Myo purported usage.[13]

Another alternative comes with recent investigation in University of Washington with gesture recognition through wireless signals [Pu13]. A strong advantage of this technology is that it does not require light-of-sight, therefore it overcomes the common problems with occlusions, as in the case of Figure 2.19 using Microsoft Kinect, with a 94% accuracy, based on the tests realized. On the other side, it is a technology still under development and presented as a proof of concept, therefore is not considered as a short-term solution [PGGP13].

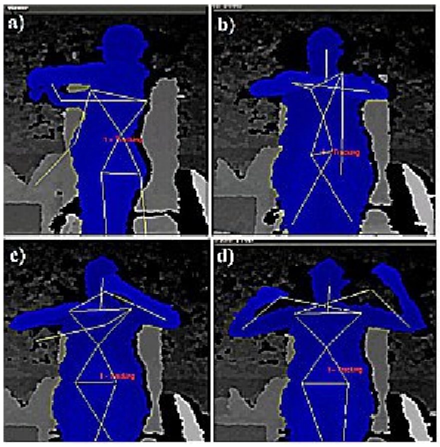

Figure 2.19: Some occlusion problems using Microsoft Kinect [ABD11].

2.6 Frameworks and Libraries

There are several frameworks and libraries used to develop gesture recognition based applications with components of natural interaction. There will be references to some that were analysed and were tested.

2.6.1 OpenNI

OpenNI is a multiplatform framework commonly used for natural interaction applications. Through this framework, it is possible to scan tridimensional scenes independently of the middleware or even the sensor used [ope13, SLC11]. In Figure 2.20, there is an example of a system using

OpenNI. Usually, OpenNI is used along with PrimeSensor NITE, because NITE allows video processing, complementing OpenNI features through computer vision algorithms for feature recognition. Besides, NITE can capture pre-defined gestures and allows the training of a set of gestures with a good recognition rate [SLC11, Pau11].

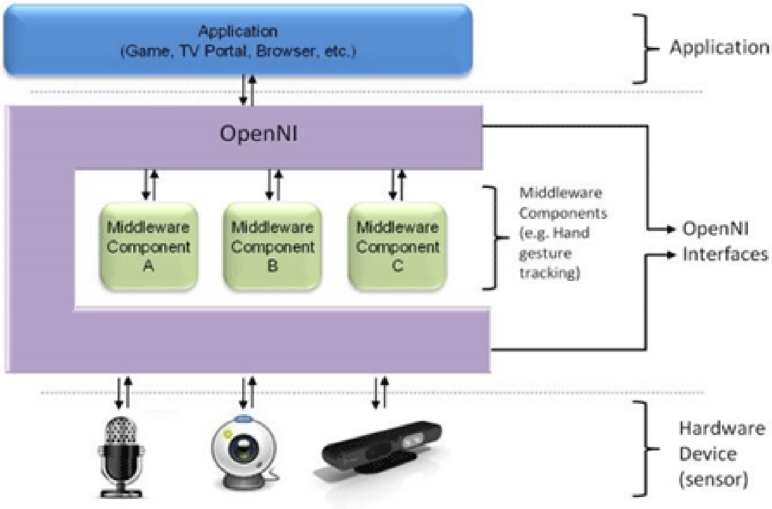

Figure 2.20: OpenNI sample architecture use [SLC11].

2.6.2 Kinect for Windows SDK

Kinect for Windows SDK is the native development kit for Microsoft Kinect. It is developed in C# and mainly works through the use of WPF services. The use of Windows Presentation Foundation (WPF) services improve the modularity of applications [M2013].

Kinect for Windows SDK allows the user to develop applications interacting with other programming languages and software, as Matlab and OpenCV algorithms, which is beneficial to improve the applications [M2013].

Another advantage is the continuous development, which is confirmed by the current version being 1.8. This development culminated with the launch of the new Microsoft Kinect, along with Xbox One in the end of 2013. The use of Kinect SDK is also an advantage as it leads to a more straightforward and organized coding [MMMM13].

2.7 Related Work

There are several projects in the gesture recognition area. However, none of the found projects include the use of cultural background to overcome the limitations of gesture recognition by gestural imitation. In these section, there is an analysis of some of the relevant projects to the state of the art of this thesis.

2.7.1 FAAST

Flexible Action and Articulated Skeleton Toolkit (Flexible Action and Articulated Skeleton Toolkit (FAAST)) is a middleware to ease the integration of body control with games and VR applications. It may use the native framework of Microsoft Kinect, Kinect for Windows SDK, or OpenNI. The main functionality of FAAST consists in the mapping of user gestures to keyboard and mouse events [SLR+11, SLR+13], which turns the interaction more natural, posing as an alternative to WIMP [FTP12, GP12]. The different approach in FAAST compared to Kinect for Windows SDK and OpenNI is that it does not require any programming skills [SLR+11] to develop applications, which turns it more accessible for any developers without programming background.

2.7.2 Wiigee

Wiigee is a Java-based recognition library for accelerometer-based gestures, specifically using Wii Remote controller. As the core of the library is written in Java, it is platform independent. This library allows the user to train his own gestures and recognize them with a high accuracy. Besides, its event-driven architecture allows the user to trigger events using gestures, which goes along with the purpose of the proposed application [PS13].

On the other hand, Wiigee is from 2008 and gesture recognition has evolved since then, so there are more recent technologies to use to complete the implementation phase, as OpenNI or using Kinect, the Kinect for Windows SDK.

2.7.3 Kinect Toolbox

Kinect Toolbox[14] is a Microsoft Kinect SDK project prepared to deal with gestures and postures. It includes a small set of gestures and it is prepared to be used as a basis for development improvements. As by the architecture of the system, each set of gestures belongs to a specific detector, which raises an event when the gesture is performed. As the solution implemented (see Chapter 3) is based on Kinect Toolkit, details will be presented on that chapter.

2.8 Conclusions

Natural user interfaces have been widely used in the past years in a broad number of contexts. Improving human-computer interaction is one of the most important topics under NUI, because the use of gestures and speech to control computers and consoles can be simultaneously appealing, but, at the same time, uncomfortable. The intention is to simplify the interaction, not harden it, so it is only meaningful to use gestures and speech to control if it effectively eases the interaction.

Nowadays, gesture recognition is a trending topic in HCI because of NI. There has been relevant evolution recently either in capturing devices with the arise of devices like Nintendo Wii, Microsoft Kinect and Leap Motion.

[14] More information on: `http://kinecttoolbox.codeplex.com/`

The various projects existent in the area include a reliable module of gesture recognition but do not include the use of the cultural background to overcome the limitations of gesture recognition by imitation. In fact, most of the applications tend to be based on interaction by imitation [CFS+10, IKK12]. The problem arises when this interaction is not possible and it is necessary to create abstractions to allow a natural and conscious interaction.

Regarding, user tracking devices, there were many choices to consider. Similar to Microsoft Kinect, Asus Xtion[15] and Primesense Sensor[16] are in the market, but, despite being very similar on the detection, the existing documentation is fewer and they are not very used in comparison.

In spite of the occlusion issues, Microsoft Kinect posed as the more valid capturing device to achieve a good result in this dissertation. The various frameworks and libraries to complement the information read by the sensors allowed improvements to the original tracking and therefore a more reliable gesture recognition system. Both of the frameworks analysed could achieve the objective of gesture recognition in real time, therefore the decision involved performance analysis and more practical tests.

Wii Remote could be an interesting alternative, but its use turns the interaction less natural because of the need to couple the device to the arm of the user restraining him.

The use of other devices, like Leap Motion and Myo, was impossible due to time constraints, as the devices are not for sale yet, in the case of Myo and because the device is very recent and lacks documentation, in the case of Leap Motion. The software is also still under development, so using one of them would pose a risk for the project as the information is still few.

[15]More information on: http://www.asus.com/Multimedia/Xtion_PRO/

[16]More information on: http://www.primesense.com/developers/get-your-sensor/

Chapter 3

Artefact

> *"It's called the shamanic interface because it was designed to be comprehensible to all people on earth, regardless of technological level or cultural background"* [Sua10].

This idea was proposed by Daniel Suarez, a computer science professional and novelist, in his novels *Daemon* and *Freedom*™. The idea uses the wide series of somatic gestures together with shamanism, showing that as even in primitive cultures there is something beyond the body - culture - to express the possibility of uniting cultural richness and gesture recognition. Therefore the idea proposed in this dissertation is an interface usable by everyone regardless of each person's cultural background, uniting culture and gesture recognition. The cultural background of the user is taken into account, on an analysis to explore the cultural variable on gestural interaction and the reduction of problems related with usability, learning curve and accessibility [Mor13].

Therefore, the proposal consisted of, as a solution and support for the following works in the area, creating an interface, with a series of predefined gestures, with cultural variations, to analyse if it helps solving the limitations of movement recognition by imitation. These gestures were defined according to the simplicity of movements and expressiveness they could achieve for a demonstration of the developed product. As some movements are not imitable, it is required to create meaningful gestural abstractions that people understand and adopt. In Figure 3.1, there is a picture from the gameplay of the mobile game Temple Run 2. If the user was controlling the character by imitation, it would be useful to have an abstraction for the character to run, because it is not practical to run long distances when playing, mainly because the usual gaming places are closed and not that spacious. As simplicity is a concern, it will only be required a camera, because any extra devices would turn the system more complex and harder to use for people without technical knowledge [CRDR11].

In this approach, besides combining people's cultural richness and gesture recognition, it is intended to simplify user's interaction with the platform by using his cultural knowledge consciously. Valli states that:

> *"simplicity leads to an easier and more sustainable relationship with media and technology"* [Val08]

which is in agreement with the objective of creating abstractions for non-imitable commands. Therefore, these abstractions may be considered simplifications of the original movement. This focus on simplicity allows a greater target public on the application, because

> *"the higher is the level of abstraction of the interface the higher is the cognitive effort required for this interaction"* [Val08].

Figure 3.1: Temple Run 2 gameplay.[1]

Then, the obvious consequences of each action become a major topic on this work as it is utterly important to create meaningful abstractions for the actions and logic consequences for them.

In sum, this was the proposal for the creation of a proof of concept, through gathering some gestures from at least two different cultures and train them in the interface. Not all gestures have meaning and there is a very broad quantity of movements that a user may do in its interaction [CRDR11]. In the system, different gestures, according to the cultural background of the user, will have the same effect, like in USAR user study [YD10]. With these, the combination of culture and gesture recognition was studied in order to guarantee abstractions for non-imitable commands.

3.1 Research Challenges

Many challenges lie in the path towards the shamanic interface. As a real-time gesture recognition system, it must recognize gestures with accuracy, not allowing confusion between gestures. According to Yin:

[1] Source: http://pulse2.com/2013/01/18/temple-run-2/

> *"Accurate gesture understanding requires reliable, real-time hand position and pose data"* [YD10].

Accurate gestures understanding is therefore the first challenge in the application, as it is essential for a solid recognition of hand movements, specially when the objective includes complex gesture recognition.

User movement tracking must be accurate, so the tracking is preferred to be realized indoors, as it is more usable for the system and because the tracking tends to be easier as conditions are generally better, as lightning and weather for example. Besides, distances are smaller and it is easier to use the tracking devices, like the ones referred in Table 2.1 [KP10]. According to Sziebig, the main objective of a tracker system is to provide “high accuracy, low latency, low jitter and robustness” [Szi09].

In this application, it is intended to use optical tracking. After an analysis of the user tracking movement devices in Section 2.5, the use of a camera arose as the best solution. It was considered important the use of devices that require the minimum technical knowledge and that are the less invasive possible for the user and the use of a camera for tracking solves that issue. There is a new approach based on wireless gesture tracking[2] that could pose an alternative to optical tracking, but at the moment it is still in development and test phase so it is not measurable the impact it could have neither on the performance of a recognition system nor in the reduction of technical knowledge to deal with the devices.

Dynamic Gesture recognition posed as a challenge. At first, in static gesture recognition using Microsoft Kinect arose the problem of occlusions. Using an improved body tracking through one of the analysed frameworks, the probability of faults could be reduced and therefore increase the accuracy of the system. For the recognition of the gestures, it was important to analyse the speed of gestures and so allow users to perform gestures with different velocities, as no user is equal to another and to improve the naturalism of gestures and HCI. However, dynamic gesture recognition poses as a more complicated challenge as it implies that the system automatically defines the beginning and the end of the gesture for a more fluid interaction [GP12]. In the created system, gestures are limited by time, because it is important not to confuse unintended movements with gestures.

Unlike [YD10], in the developed application, different users can use different gestures for the same task, depending on the user’s cultural background.

Human communication tend to aggregate two components: gesture and speech. The use of speech as a complement to gesture creates an interface more complete and effective than either component alone [KK02]. Therefore, as a complement to the work there was the development of a grammar for speech recognition of some commands to help the interaction of the user. As Krahnstoever states “to achieve natural interaction, both audio and visual modalities are fused along with feedback through a large screen display” [KK02], what corroborates the idea of combining gesture and speech recognition.

[2] More information on: `http://wisee.cs.washington.edu/`

In sum:

- In most of the situations, there are no alternatives to impossible or hard to imitate commands;
- Generally, when an abstraction exists, it is not meaningful for the user;
- The majority of NUI devices is not prepared for people with physical disabilities, because it does not provide alternatives to people with disabilities or in general commands that they can use [KBR+07, GMK+13];
- There is not a set of predefined gestures to constitute as general abstractions for HCI gestural-based devices.

3.2 Technology

3.2.1 Motion Capture Devices

As the implementation is relied on the development of a solution, it was important to select a stable technology. Microsoft Kinect was the choice between the available motion capture devices. Despite the fact that along with the launch of Xbox One, the new Kinect SDK 2.0 will be launched during 2014, the actual Microsoft Kinect has the required capabilities to detect body motion. Eventually in the future there is the possibility of adapting the system to the new Kinect, improving the system's capabilities using the new Microsoft Kinect features. As analysed on Section 2.5, none of the currently existing devices are perfect, therefore, as one of the objectives of this dissertation is simplify the interaction of the user with computers and gaming platforms, through the use of gestures, Microsoft Kinect was selected.

3.2.2 Programming Language

For the developed application, the programming language used was C#. At first there was an evaluation of the programming languages for the project, narrowing the choices to `C++` and C#. Despite the fact that `C++` is faster, C# is more straightforward and therefore more adequate for prototype applications [SA09]. It should also be noted that the use of a WPF project simplifies the task on creating a graphical interface using Microsoft Kinect as there are many contributions in the area and it works very well for a prototype.

3.2.3 Frameworks

Comparing Kinect for Windows SDK and OpenNI, the differences in the skeleton tracking can be analysed using Figure 3.2. The main difference that sustained the choice on Kinect for Windows SDK was the fact that skeletal tracking is immediate, not requiring any previous setup. As the tool developed was meant as a proof of concept and the naturalism associated to the application required the simpler tool to use to facilitate all the settings for the end user.

3.2.4 Scripting

AutoHotKey[3] is a scripting software used for task automation in Windows systems. In this project, the use of AutoHotKey allowed the use of gestures for more tasks not easily achievable without the definition of keyboard shortcuts to simplify it. Using a series of keyboard shortcut strokes, it is possible to use gestures to command the computer. The script is delivered as an executable with the application and the script file is also included to allow changes.

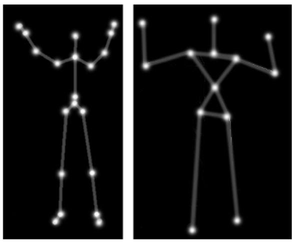

Figure 3.2: Skeleton comparison between Kinect SDK and OpenNI [XB12].

3.3 Introduction to the Developed Solution

The developed solution is included in the Design Cycle, from Design Science methodology, which is explained in Section 1.3. It uses Microsoft Kinect for the skeletal detection and consists in an implementation of a gesture recognition system, based on Kinect Toolbox 1.3.

The flow of the solution is present in Figure 3.3. After the application is running and the Microsoft Kinect is plugged in, the user places himself in the front of the capturing device and the information is sent to the Microsoft Kinect, which captures each image, calculates depth and the information of each joint. Each movement and gesture can then be performed, allowing the system to determine if the gesture is recognized. If the gesture is recognized, the system triggers the respective reaction, which affects the 3D scene included in the solution.

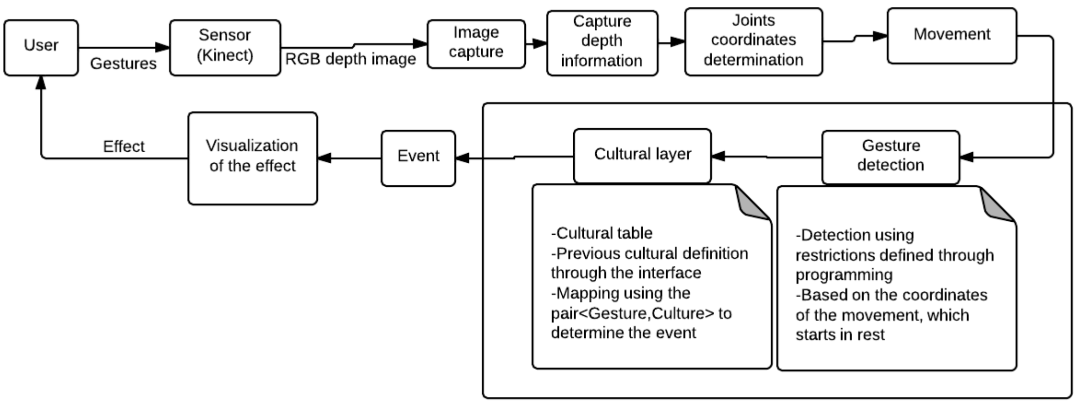

Figure 3.3: Flow of the developed solution.

[3]More information on: https://www.autohotkey.com/

3.4 Architecture and Gesture Recognition

One of the main purposes of this thesis relied on creating an implicit activity-driven interaction system, as the ones referred by Lukowicz [LARC10]: an environment where gestures and movements are meaningful and the result is therefore obvious for the user.

As in this thesis, some games "match specific in-game actions, such as walking, with similar real world movements" [Cha07] and in this case, the application creates a meaningful abstraction for the user to use conveniently.

As the tool is intended to be offered to a wide audience, it is important to match certain criteria and then it urges the fact that elderly people have a notorious decrease in acuity, memory and attention [Cha07]. It is also relevant the reduction of muscle mass, because all these factors must be taken into account when building the application. To support people in a wheelchair or with movement restrictions, a seated mode is allowed in the developed application.

The simple scene created for demonstration purposes was done using the Tao Framework [4]. This framework allowed the use of OpenGL in the project, under a .NET environment. Despite having some alternative, this was successful as it allowed to receive keystrokes from another applications, as it is done with the developed application for the gestural interaction to affect the scene. Another alternatives have been tested, such as SharpGL[5], but despite a better looks, unfortunately it did not accept the reception of keystrokes from external applications.

Regarding the implementation, the flow of the system is defined in Figure 3.3. This implementation is based on Kinect Toolbox 1.3, which is a set of useful tools, including some gestures and postures that allow gestural interaction with an application. Microsoft Kinect captures a data stream, sending it through USB to the computer, allowing its visualization in real time. This data is gathered by the KinectSensor class, which allows the detection by Gesture Detector class and its descendants (see Listing 3.1). The data is converted from the sensor to a two system coordinate to be presented to the user in the screen.

```
SwipeGestureDetector swipeGestureRecognizer;

private void Initialize()
    {
    //a code excerpt to explain the use of gesture detectors, provided by
        Kinect Toolbox
    swipeGestureRecognizer = new SwipeGestureDetector();
    swipeGestureRecognizer.OnGestureDetected += OnGestureDetected;

    kinectSensor.Start();
    }
```

Listing 3.1: The use of detectors in the developed application.

[4]More information on: http://sourceforge.net/projects/taoframework/
[5]More information on: http://sharpgl.codeplex.com/

The determination of the gesture is a process based on different gesture detectors. Kinect Toolkit included some in its architecture, which are represented in Figure 3.4. The ones shaded are the ones which include major changes for the approach used in this work. The objective of these detectors is separate detectors according to the similarity of gestures. Therefore, each detector includes various characteristics relevant for each type of gesture. The evolution of the system would require the creation of more subclasses of GestureDetector if gestures were linear. If gestures were complex, it would require the use of additional classes based on Kinect Toolkit architecture.

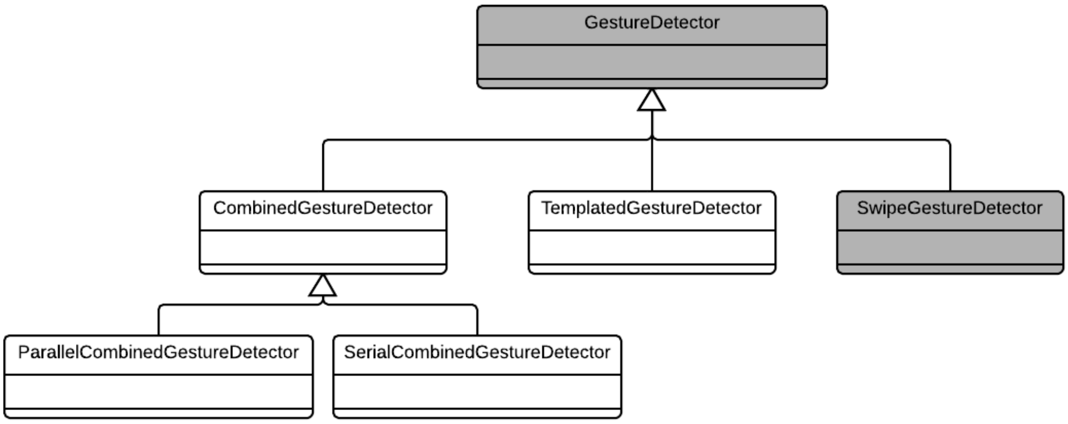

Figure 3.4: Gesture Detector diagram.

After gesture detectors are registered, they are passive and waiting for the user to perform any gesture. Each detector has its own attributes, specific and adequate to the gestures they are prepared to recognize. The analysis of each movement is based on restrictions upon the points where the gesture takes part. This means that if the gesture satisfies the conditions, it is accepted by the detector and the respective feedback is expressed. An example for a SwipeToRight is expressed in Listing 3.2.

```
if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) < SwipeMaximalHeight, // Height
                (p1, p2) => p2.X - p1.X > -0.01f, // Progression to right
                (p1, p2) => Math.Abs(p2.X - p1.X) > SwipeMinimalLength, // Length
                SwipeMinimalDuration, SwipeMaximalDuration) // Duration
                )
```

Listing 3.2: Determination of a movement through restrictions.

Analysing SwipeGestureDetector attributes, it is possible to see which characteristics are relevant for the recognition of the gesture (see Listing 3.3).

```
  //Swipe detector specific attributes
    public float SwipeMinimalLength { get; set; }
    public float SwipeMaximalHeight { get; set; }
```

```
public int SwipeMinimalDuration { get; set; }
public int SwipeMaximalDuration { get; set; }
public float SwipeLength { get; set; }
```

Listing 3.3: Proper detector characteristics.

- SwipeMinimalLength expresses the minimal value to consider the swipe;
- SwipeMaximalHeight expresses the maximal variation value on the movement on other coordinates: in case it is intended to do a SwipeToRight, if the movement has a great variation on Y coordinate, the gesture will not be recognized;
- SwipeMinimalDuration and SwipeMaximalDuration express the time restrictions of the movement. Not to allow too abrupt movements or too slow. This time interval permits that different users perform the gestures at their own pace, because not all users take equal time performing the same gestures;
- SwipeLength represents the minimum distance needed in the intended coordinate to be considered a Swipe.

The amplitude of the movement can be taken into account producing different outputs on the system. This functionality can be considered as a filter on the movements because it depends on general conditions and available to all the detectors.

Having feedbacks being outputted correctly, the cultural layer was introduced in between. The idea of creating a lookup table was decided. Given a pair constituted by the gesture's name and a Culture, it could get the feedback of the application, which can be seen in the excerpt of code, Listing 3.4.

```
public Dictionary<Tuple<string, Culture>, string> CulturalMapping = new
    Dictionary<Tuple<string, Culture>, string>();
```

Listing 3.4: Cultural Mapping Lookup table definition.

The user moves freely towards the camera, performing different movements. Each one of the detectors includes a LookForGesture() method that gets the selected value of the Culture from the interface. When a gesture is recognized, an event is then launched to the cultural layer. After that, it analyses the positions of the specific joints during the movement according to the conditions defined in Listing 3.5.

```
protected override void LookForGesture()
  {

  Culture selectedItem = (Culture)App.win.CultureCombobox.SelectedItem;

```

```
//if the gesture is a SwipeToRight
if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) < SwipeMaximalHeight, // Height
            (p1, p2) => p2.X - p1.X > -0.01f, // Progression to right
            (p1, p2) => Math.Abs(p2.X - p1.X) > SwipeMinimalLength, // Length
            SwipeMinimalDuration, SwipeMaximalDuration) // Duration
            )
    //action taken when the gesture corresponds
    foreach (KeyValuePair<Tuple<string, Culture>, string> pair in CulturalMapping)
            {
                if (pair.Key.Equals(Tuple.Create("SwipeToRight", selectedItem))){
                    RaiseGestureDetected("SwipeToRight");
                    System.Windows.Forms.SendKeys.SendWait(pair.Value);
                    //System.Windows.Forms.SendKeys.Send(pair.Value);
                    Console.WriteLine(pair.Value);}}
```

Listing 3.5: Look for gesture example.

The developed interface can be observed on Figure 3.5.

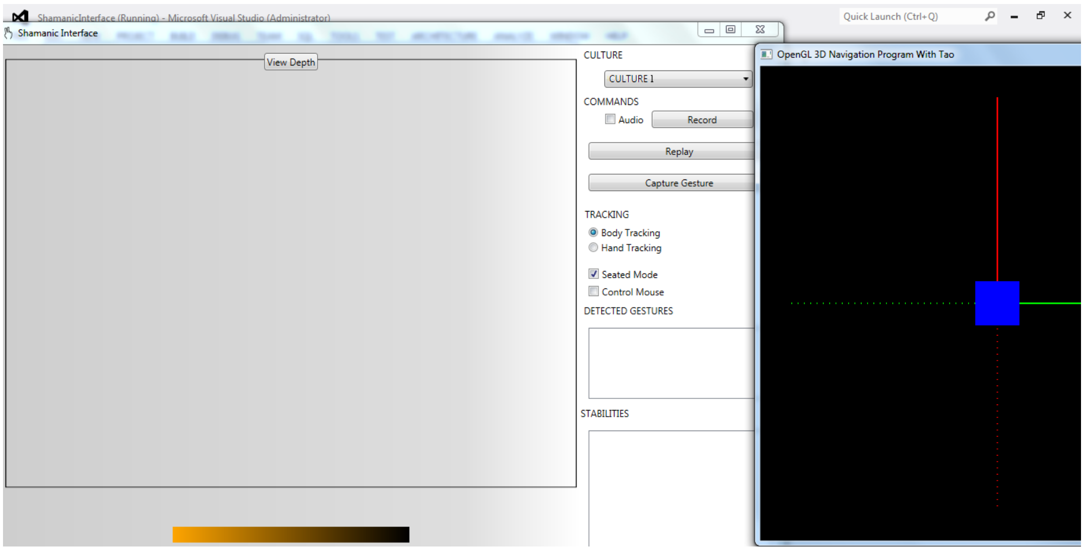

Figure 3.5: Interface of the gesture recognition system.

In the cultural layer, there is a search in the lookup table for the defined pair (represented as a Tuple), composed by the gesture and the culture, to retrieve the correct consequence. This consequence is then automatically activated, affecting the scene (see on Figure 3.6). The effect is based on keystrokes, to more easily map consequences on different applications.

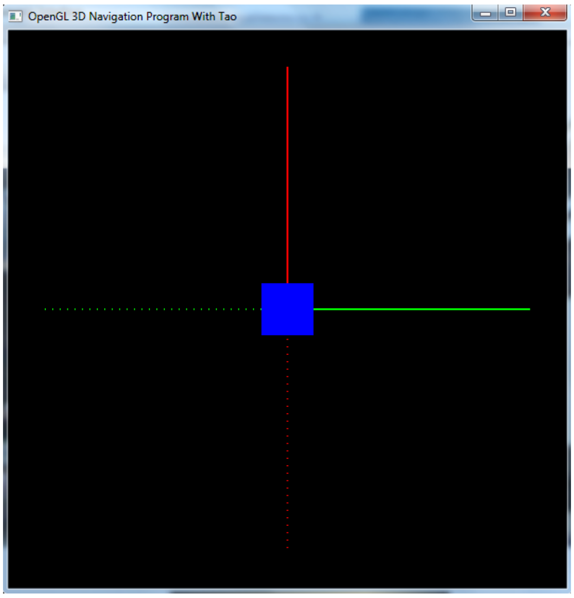

Figure 3.6: Scene defined to evaluate the outputs of the application.

3.5 Culture Integration

The importance of the cultural background of an individual to understand his interaction with a system is referred in the work from [RBA08]. It proposes the analysis and inclusion of this variable to be taken into account in the design of interfaces.

Several questions rose in this work regarding culture integration. One of them was certainly how to include culture to check if it was measurable or deductible. As that analysis would require a heavy computation and time spent, it is required for the user to introduce its cultural background when initializing the application. This saves time and reduces (or even eliminates) analysis error regarding the cultural background of each user.

The use of culture as a computational notion is important and in this application tackles the way the user interacts with it. The configuration of different cultures can be compared to a translator change. Meaning that user's with different cultural backgrounds may expect different outcomes from each recognised gesture, depending on their history, as expressed on the following code excerpt. The detailed mapping is expressed on Table 3.1.

```
    CulturalMapping.Add(Tuple.Create("SwipeToRight", Culture.Culture1),"{Right}");
    CulturalMapping.Add(Tuple.Create("SwipeToRight", Culture.Culture2), "{Down}");

    CulturalMapping.Add(Tuple.Create("SwipeToFront", Culture.Culture2), "{Right}");

    CulturalMapping.Add(Tuple.Create("SwipeToLeft", Culture.Culture1), "{Left}");
    CulturalMapping.Add(Tuple.Create("SwipeToLeft", Culture.Culture2), "{Left}");
    CulturalMapping.Add(Tuple.Create("SwipeToLeft", Culture.Culture3), "{Left}");
```

Listing 3.6: Some of the values in the Lookup table.

Table 3.1: Cultural Gesture Mapping table definition. This data was used for evaluation.

Gesture	**Culture**	**Feedback**
SwipeToRight	Culture1	Right
SwipeToRight	Culture2	**Down**
SwipeToRight	Culture3	Right
SwipeToLeft	Culture1	Left
SwipeToLeft	Culture2	Left
SwipeToLeft	Culture3	Left
SwipeToTop	Culture1	Up
SwipeToTop	Culture2	Up
SwipeToTop	Culture3	Up
SwipeToDown	Culture1	Down
SwipeToDown	Culture2	Down
SwipeToDown	Culture3	Down
SwipeToFront	Culture1	PGUP
SwipeToFront	Culture2	**Right**
SwipeToFront	Culture3	PGUP
SwipeToBack	Culture1	PGDN
SwipeToBack	Culture2	**Left**
SwipeToBack	Culture3	PGDN

Besides the use of dynamic gesture, such as the ones expressed by Table 3.1, there was also the definition of some predefined postures, based on previous existent work from Kinect Toolbox. The respective cultural mapping was done as presented in Table 3.2.

Table 3.2: Cultural Posture Mapping table definition.

Gesture	**Culture**	**Feedback**
HandsJoined	Culture1	Down
HandsJoined	Culture2	**Up**
HandsJoined	Culture3	Down
LeftHandOverHead	Culture1	Up
LeftHandOverHead	Culture2	Up
LeftHandOverHead	Culture3	**Left**
RightHandOverHead	Culture1	Up
RightHandOverHead	Culture2	Up
RightHandOverHead	Culture3	**Right**

The values in bold from both Tables 3.1 and 3.2 are the ones relevant for testing issues. The inclusion of some of these values in the Dictionary used for the mapping is expressed in Listing 3.6.

CulturalMapping is a C# dictionary composed by a Tuple and a string. The Tuple is composed by a string - the gesture - and a Cultural element, which are represented in Listing 3.7.

```
    public enum Culture{
        [Display(Name = "Culture1")]
```

```
        [Description("Culture1 gestures.")]
        Culture1,
        [Display(Name = "Culture2")]
        [Description("Culture2 gestures.")]
        Culture2,
        [Display(Name = "Culture3")]
        [Description("Culture3 gestures.")]
        Culture3}
```

Listing 3.7: Enumeration Culture.

The value from the dictionary is a string, which represents the command sent to the foreground application. This command represents a keystroke sent using SendWait function.

The recognition of a gesture is represented in Listing 3.8.

```
    if (ScanPositions((p1, p2) => Math.Abs(p2.Y - p1.Y) < SwipeMaximalHeight, //
        Height
                (p1, p2) => p2.X - p1.X > -0.01f, // Progression to right
                (p1, p2) => Math.Abs(p2.X - p1.X) > SwipeMinimalLength, // Length
                SwipeMinimalDuration, SwipeMaximalDuration)) // Duration
        foreach (KeyValuePair<Tuple<string, Culture>, string> pair in
            CulturalMapping){
                    if (pair.Key.Equals(Tuple.Create("SwipeToRight",
                        selectedItem))){
                        RaiseGestureDetected("SwipeToRight");
                        System.Windows.Forms.SendKeys.SendWait(pair.Value);
                        Console.WriteLine(pair.Value);}}
```

Listing 3.8: The SendWait method used after positive recognition of SwipeToRight.

Observing the values on the lookup table, there are various relevant cases for our application: cases when a movement has the same output in each defined culture, cases when the same movement has different meanings according to a culture and cases when two different movements have the same output, each one from a different culture. This gathered data for testing purposes proved very useful to determine the results of the implementation.

3.6 Applications

There are several possible applications for the implemented system. The creating of a scene is purely for demonstration purposes and to be adapted in future versions of the product.

The actual application allows the control of other applications in foreground, using the SendWait method from C#, which allows to send keystrokes to applications. It is therefore dependant on the correct mapping of keys and actions from other applications. The use of AutoHotKey permits customization on that matter.

As a proof of concept, this system needs future improvements to allow a better use on their targets someday. Augmented Reality applications, games and the generic control of computer and gaming platforms are some of the applications intended for this gesture recognition system.

3.7 Difficulties

One of the major handicaps is the low number of skeleton points tracked. The limitation to 20 tracked points invalidates a full-body precise tracking along with hand tracking. This poses as a major difficulty to create a gestural controller system. To overcome this challenge, there has been the idea of use tracking in different moments: at one moment, the tracking is fully concentrated on the hand skeleton and motion and at another, the tracking concentrates on the full-body without special focus on hands. At the moment, due to schedule restrictions, the hand tracking is not fully implemented and functional.

The limitations regarding the use of OpenGL in C# were also complicated to handle because it was difficult to find a good framework that received keystrokes from another applications and therefore acted accordingly.

For last, the fact that the area is so unexplored was a handicap because there were many paths to follow, creating a wide range of options to define the work to perform.

Artefact

Chapter 4

Evaluation

According to [HBC+96]:

> Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

Using this definition, some metrics can be defined to analyse the proposed solution for the problem. As this is a HCI system, the system's design and usability must be taken into account, but the major concern is the performance of the application. As a natural interaction based system, it is also expected to have a good response-time, because it is a real-time application.

Evaluation, as in any software application, is an important component to assess the quality of the product. In this thesis, the guideline, as referred in Section 1.3 is Design Science, which is expressed in Figure 1.7.

The analysis of the results was done according to the Design Science metrics (presented on Table 1.2) on various parameters, as usability, functionalities and improvements comparing to the state of the art applications.

4.1 Testing Results

4.1.1 Methodology

The methodology followed during this dissertation in was Design Science, which is detailed in Section 1.3.

As explained above, different evaluation scenarios can be used to support this work, but due to time constraints evaluation is not as detailed as intended for scientific projects. Following this methods, expressed in Table 1.2, there was much work involved on the validation of this thesis. Regarding analytical methods, it was performed a statical analysis. Despite gesture complexity is not high, the combination of gestures and precise detection is very important. Dynamically speaking, the performance of the system is great, because the response is instantaneous. This is a very important topic, because a gesture recognition system, it must be responsive as users expect fast results and, in this case, instantaneous feedback.

Controlled experiences and simulations are included in the experimental methods. These experiences, together with the Testing methods, include all the testing process involving the solution developed during this dissertation. These are detailed in the Subsection 4.1.2.

For last, descriptive methods include all the data gathered during Literature Review that support the importance of the system as a first step in the evolution of the concept of the shamanic interface, therefore utterly important for future developments in the area.

4.1.2 Proof of concept

Regarding the developed implementation, it is worth note that all the details of the solution are explained in Chapter 3. The designations of cultures are a start for future improvement, but at this time, the defined gestures have no specific connection to the names defined.

As any software tool, during development, tests are always present to assure the correct functionality of the system. Therefore, the first step on this approach is related to assure the correct detection of the human skeleton, taking into account Microsoft Kinect limitations. The detection using the application is expressed on Figure 4.1.

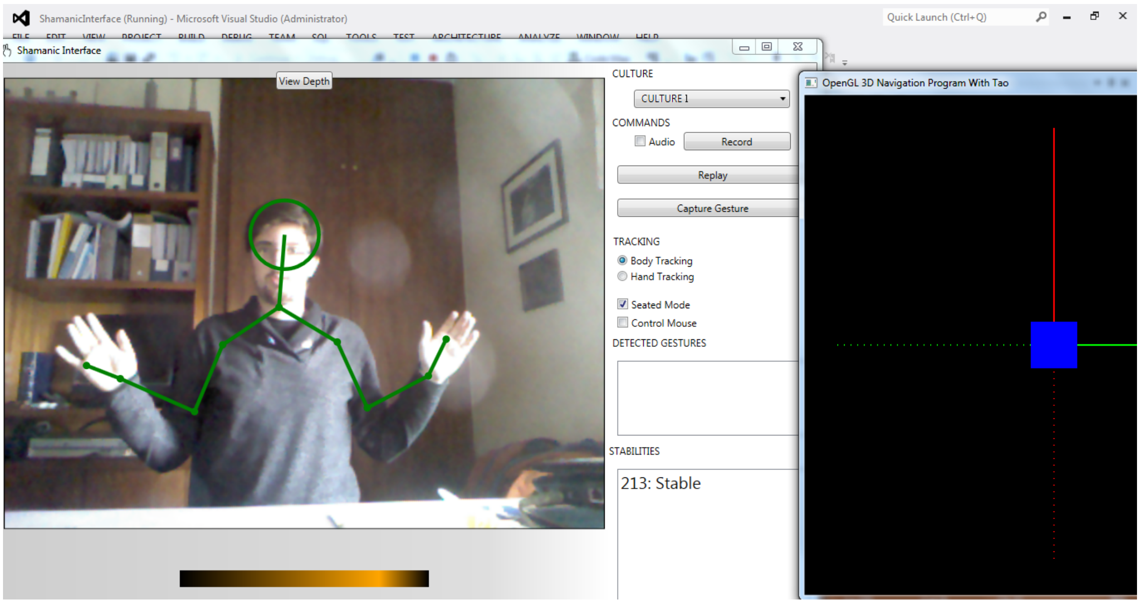

Figure 4.1: Skeletal tracking using Kinect in seated mode on Shamanic Interface application.

Following, the objective of adding different gestures and detectors is important, so it is important to understand the scope of each one of the already defined gestures and deeply test the gestures newly created. Same logic for postures, to understand their limits and eventual Microsoft Kinect failures related to them. This testing program required some time to allow the detection to be the more precise possible.

At the time, the idea of creating specific feedback was not well precise, so actions produced the same feedback: a simple message box specific to each gesture to express the correct completion of the gesture. This approach was useful in the beginning to test if feedbacks were given correctly, depending on the performed gesture.

The following set of tests included the correct inclusion of this layer and the verification of a correct mapping according to defined gestures in the application.

After this setting of tests, the intuitive and obvious cultural feedback was missing. The creation of a scene to observe gestural effects on a cube was a simple solution to this problem. Due to time constraints, a more elaborated scene was not completed to allow different and more complex interactions, but that is expected to be worked on in the future.

On this validation phase, tests consisted on different and obvious effects according to the culture.

In Figures 4.2 and 4.3, we can observe the different parts of the movement SwapToRight under Culture 1, which presents a movement of the horizontal referential to the right. The complete movement is presented as a video in http://www.youtube.com/watch?v=J56aBvF6gW0.

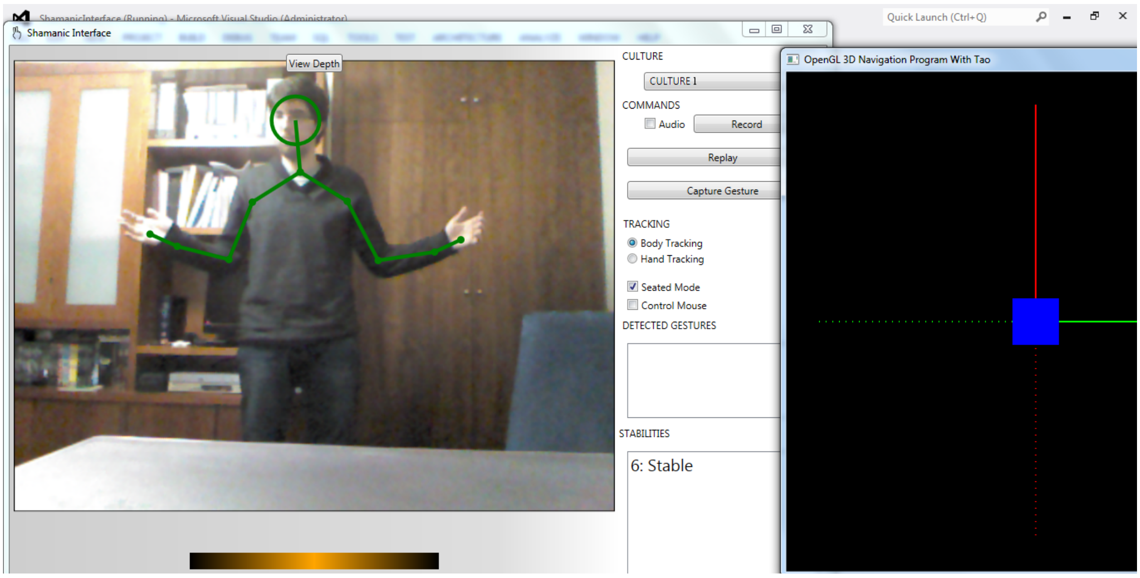

Figure 4.2: Beginning of the Swap to the Right movement.

On the other hand, under Culture 2, the same movement provides a different feedback, as observed in Figure 4.4. This situation is also expressed in a small video presented in http://www.youtube.com/watch?v=WPOtXhvYTj0 to facilitate the understanding of the movement.

It is also important to note the importance of testing gestures that return the same feedback in different cultures, which is expressed in Figure 4.5 and in http://www.youtube.com/watch?v=72ww-MSNErI. In this case, the movement SwapToFront in Culture 2 has the same result as SwapToRight in Culture 1.

This tests were important to analyse the feasibility of the integration of cultural background in a gesture recognition system and also the more important features to focus, specially for future developments in the area.

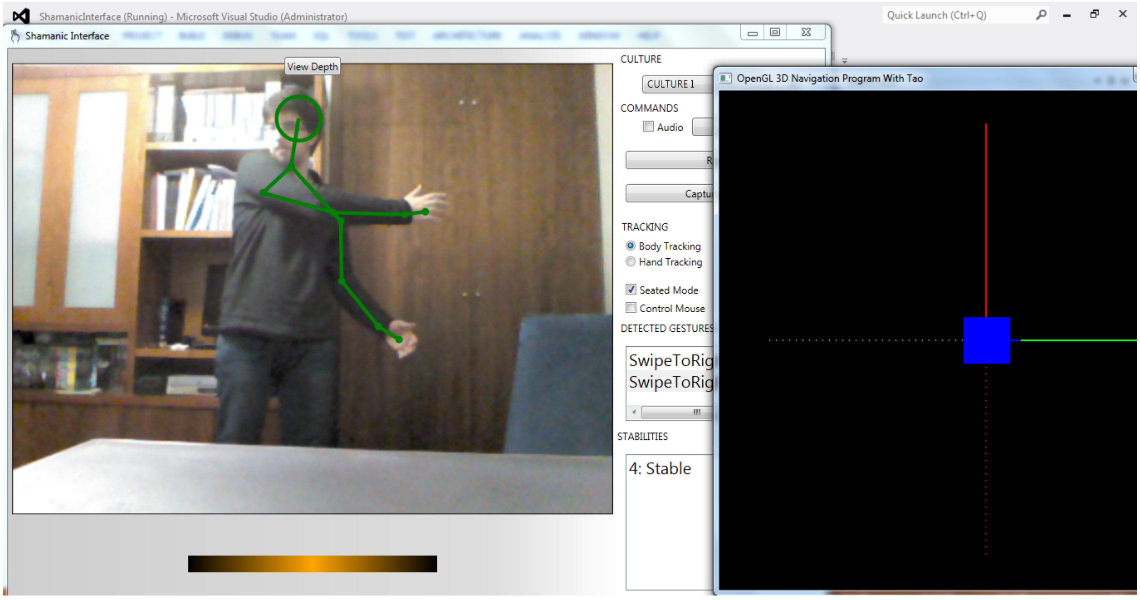

Figure 4.3: The end of the Swap to the Right movement.

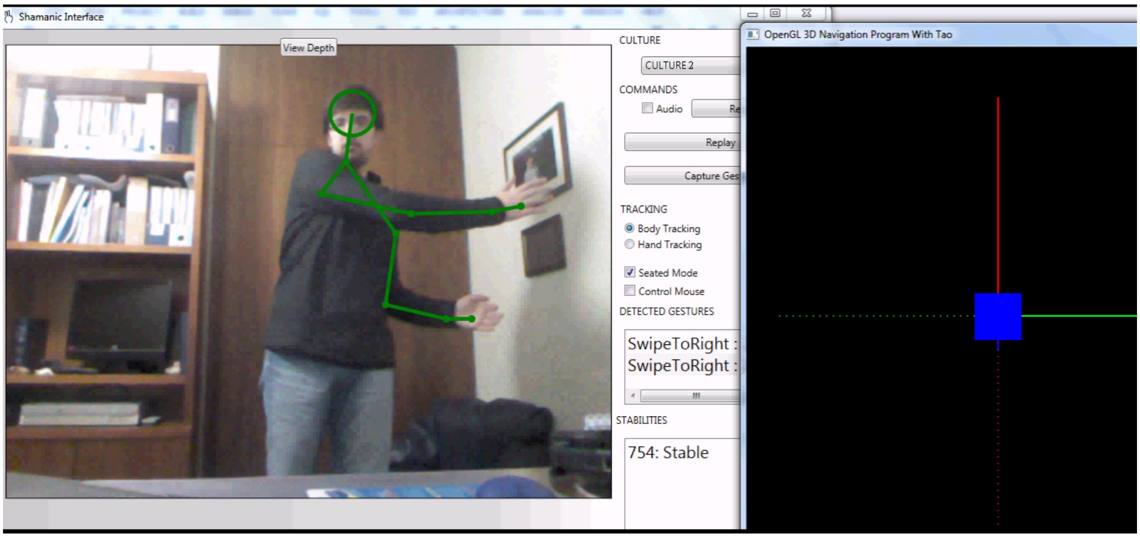

Figure 4.4: Swap to the Right under Culture 2.

The process of selection of more specific cultural feedback would require further studies and eventually the testing by future users of the system to be accurate. Due to time constraints on this dissertation, this situation was impossible, but it is certainly an interesting point for future evolution of the system.

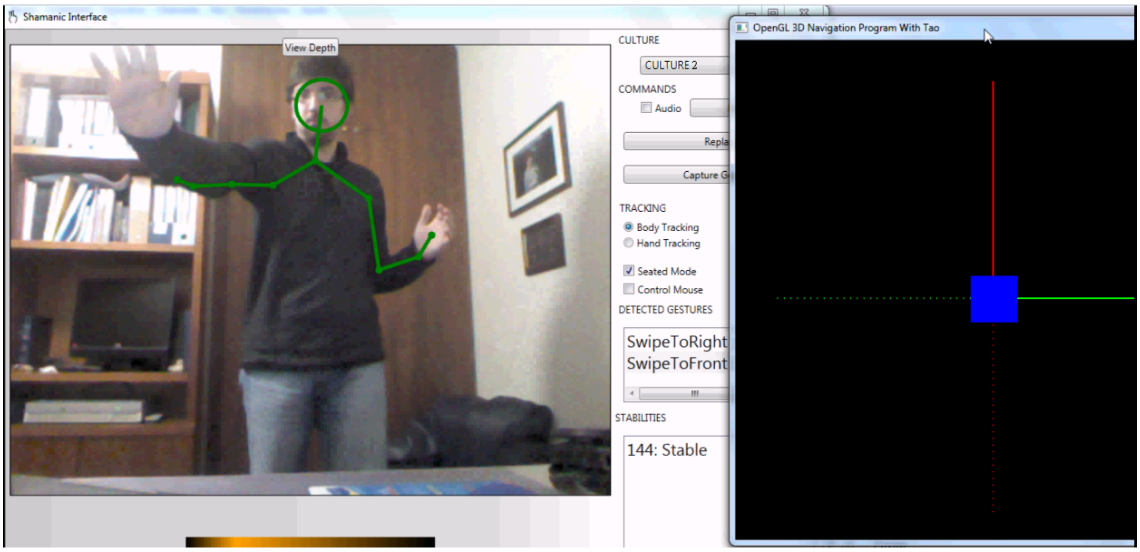

Figure 4.5: Swap to Front under Culture 2.

4.2 Discussion

After a testing phase it is important to discuss the results and trace the future of the application and specify the next steps on evaluating it. Not all the intended evaluating methods were applied because of time restrictions involved on this dissertation, but the plan is traced to allow a more deep and fruitful evaluation in the future. This plan will go according to the methodology defined during this dissertation, Design Science.

Technically speaking, one of the next phases of the project involves the migration from Microsoft Kinect SDK to Microsoft Kinect SDK 2.0, which will be going public in the Summer 2014. Therefore, technical improvements on the tracking will be mainly depending on the improvements associated to the new sensor functionalities and require a significant wait. Meanwhile, it is essential to reinforce the importance of the developed solution to a wider public.

To prove the importance of the developed gesture recognition system, some more gestures culturally related should be defined to provide substantial data for user testing. The relevance of user testing would show the acceptance of the defined gestures and also from the developed solution in some end-users. This data would also be important to gather information on how to improve the system in the user point of view.

Regarding other secondary objectives of this project in long term, it would be interesting to evaluate the system using real cultural gestures. Besides usability analysis and tests with physically limited users would be a good complement, because it would show if the system is really prepared to deal with physical limitations (the inclusion of a seated mode is surely not enough) and it would provide suggestions to implement in the future to allow a more inclusive solution.

The filter functionality also requires some improvements to allow different measures on the movement, such as rhythm, for example. The feedback from users can be important to evaluate the incorporation of new characteristics on the filter.

4.3 Conclusions

The obtained results were enlightening about the positive evolution on the use of the interface. The interface tended to become more intuitive and gestures became more objective and the respective feedback became more predictable. Obviously, this results need much more testing with different individuals and using real cultural gestures, but after this first step, many paths lie ahead in the future of cultural-based gestural recognition.

Chapter 5

Conclusions and Future Work

This chapter presents an analysis of objective satisfaction, main contributions and directions for future developments in the area.

5.1 Objective Satisfaction

After analysing the state of the art, it is possible to conclude that communication through gestures is an innovative and trending topic in human-computer interaction. The emergence of new devices like Myo[1] and Leap Motion[2] proves this point. Besides, even new types of interaction are arising, such as WiSee[3], which preliminary results show very good accuracy - 94% - in wireless gesture recognition [PGGP13].

Between the capturing devices analysed, Microsoft Kinect and Nintendo Wii Remote are the most consistent choices to receive the input from the user, as they are very well documented and the gesture recognition is eased, because these devices provide additional functionalities, such as the infrared depth camera (Microsoft Kinect) and the accelerometer (Nintendo Wiimote). Besides, both are mature devices and therefore, stable for future development.

In this area, Kinect gains advantage by not needing to couple any device with the user, as the Wiimote needs to track the movements of the arm through the accelerometer [SPHB08]. Kinect only requires the use of a camera for tracking the movements of the user, becoming a less invasive approach. This is important because the simpler the system is, the more users will use it. On top of that, a system that does not need any technical knowledge becomes even more relevant to turn a system more user-friendly [GP12]. Despite that fact, user tracking using Kinect must suffer some improvements to allow consistent gesture recognition, due to occlusions [ABD11].

Another important issue is the launch of the new Kinect with the majority of the existent limitations solved in the beginning of 2014 [Hed13]. In the end of the implementation, it is important to analyse the retroaction of Kinect applications to allow the system to take advantage of a new capture device and last longer. Clearly the use of the new Kinect would be a major

[1] More information on: https://www.thalmic.com/myo/
[2] More information on: https://www.leapmotion.com/
[3] More information on: http://wisee.cs.washington.edu/

improvement to this project, but as it will only predictably be launched in 2014, it is important for future improvements.

It is also expected that sensing systems evolve to ease the recognition and interpretation of complex human movements and not only less complex gestures[LARC10].

Several reflections rose from this dissertation. At first, the relevance of the studied topic. The area is greatly unexplored, leaving room for several improvements in upcoming years. Some of the proposed improvements are explained in the Section 5.2.1.

Regarding objective satisfaction, the scope of the shamanic interface is too wide to fulfil during a master thesis. Although, the objectives proposed can be seen as completed, because for this dissertation, the objectives were to provide a basic solution on this idea and support future works on the area.

Remembering, from Chapter 1, the expected contributions:

- How are gestures mapped into actions?
- What perspectives are relevant in gesture analysis?
- Are there gestures that can constitute abstractions for human-computer interaction based on gestures when commands are hard or impossible to imitate?
- Are they meaningful?
- Is it possible to integrate culture to consider this interaction meaningful?

As explained in detail in Chapter 3, gestures are mapped into actions, using a lookup table, in the proof of concept, more specifically, a C# dictionary. Therefore, from a gesture and a culture combined, using the lookup table, the user gets the expected feedback.

Gesture analysis includes several points as seen in the Literature Review in Chapter 2. In this area, the main gesture recognition systems tend to depend heavily on gesture training to increase the number of gestures detected. On the other hand, on the developed system that feature was not planned, although it would be a major addition on an eventual future release. To recognize gestures, usually systems tend to recognize simple gestures without adding different perspectives on them. Gestures can be different one another, depending on amplitude of the movements and rhythm for example. To overcome this issue, a filter layer was created, separating movements also according to amplitude and rhythm, originating different feedbacks.

The analysis and testing of the system was continuous and always looking for logical feedbacks for the defined gestures. This work can be improved, using data from future users and through a more deep analysis of different cultures and habits. According to the way the mapping is achieved, the substitution of difficult to perform or impossible gestures is possible, changing the feedback of the system. In the Section 5.2.1, there are several references to a way of expressing this more naturally using Rinions[4], Microsoft Kinect and OpenSimulator[5]. There would be other

[4]More information on: http://www.nsl.tuis.ac.jp/xoops/modules/xpwiki/?Rinions

[5]More information on: http://opensimulator.org/wiki/Main_Page

platforms and technologies to achieve this idea, but this one was briefly studied and could be a major update to the developed system. Despite gestures are not culturally related at the time, the used feedbacks are simple and intuitive to explain the concept.

The use of Microsoft Kinect for this work was a good choice and the possibility of migrating the system to Microsoft Kinect SDK 2.0 is a major advantage as it is expected that the proof of concept can combine hand and body detection at the same time with precision. On the other way, at the moment, gesture detection using Microsoft Kinect SDK tends to depend on firing events when the gestures are detected. Using this approach, a scability issue may arise when the number of detectors and fired events is too high, but it is worth note that no issues surged during the development.

5.2 Future Work

5.2.1 Improvements

Gesture recognition has been shown to be very useful in HCI. Nevertheless, there are inherent limitations related to performing some movements, as some are difficult to perform and others are impossible to replicate. Besides, the impact of the inclusion of the cultural component to overcome the challenges of creating meaningful gestural abstractions as alternative movements is not well studied.

As future work, there are many paths ahead on this research.

At first, for a better gesture interaction system, hand motion would be an interesting add-on, specially related to static gestures. At the moment, the only possibility to have hand tracking along with body tracking is to divide on phases, because 20 tracking points are not enough for a precise simultaneous hand and body tracking.

Along with the launch of a new Kinect, there will be a new SDK which is expected to allow a more precise detection of hand and body. It is also expected that the tracked points are increased, which would go according to the idea intended in this work: the Shamanic Interface would benefit from more points to track, allowing the simultaneous precise detection of hand and body motion. Along with the arise of the new version of the Kinect device, it is required the adaptation of the proof of concept to the new version of Kinect, using the new features and improving the system substantially. According to Microsoft Kinect 2.0 developers, code adaptations are not very hard, so it should not be difficult or that much time consuming to adapt the solution.

Being this implementation a proof of concept, it is not really complete in terms of gesture diversity. It would be an improvement the addition of more complex gestures and a broader set of gestures, also to allow a more significant testing in the evaluation phase. Therefore, a more wider evaluation phase, including different users, preferably from different cultural backgrounds. This diversity of testers would enrich the system and provide valuable feedback to help understand if the system is evolving in the right path. Besides, suggestions of improvements on the mapped gestures and association to real cultures are the logical step of a system like this.

Through all the recent research in HCI through gestures, new devices like Myo and the new version of Kinect are being launched in the next months. After the launch it is interesting to analyse the impact of integrating any other devices with Kinect, improving the control of computers and electronic devices. As the new Kinect will substitute the actual version, this could be done using Myo, but keeping in mind that one of the important points of the investigation addresses the simplification of the interaction so using coupled devices such as Myo can turn this situation more difficult. This was also one of the reasons Nintendo Wii Remote was excluded from the study, but it is expected that Myo bring certain advantages and less discomfort than the Wiimote and also more precision and diversity of movements, because it does not only depend on an accelerometer as Nintendo Wii Remote.

Combination of devices could be further analysed. Despite the possible improvements, the study during this dissertation concluded that the complexity of the system would substantially increase, therefore not posing as a natural alternative to interact with computer and gaming platforms.

The use of a scripting utility such as AutoHotKey allowed the creating of an executable with simple, not used by the system short-keys to use in the application. Therefore, this executable can be used by any user and he can change the script and generate it again. Another improvement would be turn this task more automated, using forms in the project in C# language to change this information without requiring much technical knowledge.

Actually, the use of emulated keystrokes allows a great variety of usages, allowing the user to interact with another applications, as long as they are in foreground. Even though, it is very difficult to control a variety of applications, defining interaction models could be a very interesting step, allowing similar applications to behave equally when the same keystrokes are received. This homogeneity would be very useful for a real scalable application, relating to other applications control. The possibility of adding keypress actions is also very interesting as it would allow a more precise control of applications, not being restricted to keystrokes.

Furthermore, the use of avatars to express the real movements the user pretends when he does the alternate movements would be a major benefit to disclose the proposal. It would simultaneously help people understand more uses of the application and also simulate better its final usage. A way to realise this addition would be using Rinions to make the communication between Microsoft Kinect and OpenSimulator. After the communication is achieved, the movements realised by the user are recognised by the system and then the pretended movement is executed by the avatar. This could be an addition on a different mode of the application. The skeletal tracking is already done using Microsoft SDK and presented to the user, so this addition would complete the information provided by sending commands to move an avatar using for example OpenSimulator as referred. The use of avatars to express movement would increase the potential of the application and also its applicabilities, such as in personal games using virtual characters for example.

In sum, this area is wide and is not very explored, therefore the progression possibilities are strong and many different paths lie ahead to increase the knowledge in the area.

References

[ABD11] Isabele A Agostinho, José Remo Ferreira Brega, and Diego R Colombo Dias. Combinação de Dispositivos de Baixo Custo para Rastreamento de Gestos. Technical report, 2011.

[BB11] K. K. Biswas and Saurav Kumar Basu. Gesture recognition using Microsoft Kinect®. In *5th International Conference on Automation, Robotics and Applications*, volume 2, pages 100–103, Wellington, New Zealand, December 2011. IEEE.

[BFR10] Hans-joachim Bieg, Roland W Fleming, and Harald Reiterer. Eye and pointer coordination in search and selection tasks. *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*, page 89, 2010.

[CFA+10] Julie Carmigniani, Borko Furht, Marco Anisetti, Paolo Ceravolo, Ernesto Damiani, and Misa Ivkovic. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51(1):341–377, December 2010.

[CFM+13] Fernando Cassola, Benjamim Fonseca, Leonel Morgado, Paulo Martins, Hugo Paredes, and Fausto de Carvalho. Online Gym : um ginásio virtual 3D integrando a Kinect – análise comparativa de bibliotecas de suporte. 2013.

[CFS+10] Sylvain Calinon, D Florent, Eric L Sauser, Darwin G Caldwell, and Aude G. Billard. Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression. *IEEE Robotics and Automation Magazine*, 17(2):44–54, 2010.

[Cha07] Adam S Champy. Elements of Motion : 3D Sensors in Intuitive Game Design. *Analog Dialogue*, 41(04):1–4, 2007.

[CRDR11] Ankit Chaudhary, J.L Raheja, Karen Das, and Sonia Raheja. Intelligent Approaches to interact with Machines using Hand Gesture Recognition in Natural way: A Survey. *International Journal of Computer Science & Engineering Survey*, 2(1):122–133, February 2011.

[DB08] Henry Been-Lirn Duh and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202, Cambridge, UK, September 2008. IEEE.

[DMKK12] Hipólito Douglas, França Moreira, Claudio Kirner, and Tereza Gonçalves Kirner. Desenvolvimento de Quiosque para Interação em Aplicações de Realidade Virtual Utilizando Dispositivos de Interação Natural. Technical report, 2012.

[Eps13] Zach Epstein. Microsoft Xbox 360 sales. http://bgr.com/2013/02/12/microsoft-xbox-360-sales-2013-325481/, accessed in July 2013.

[FTP12] Rita Francese, Genoveffa Tortora, and Ignazio Passero. Wiimote and Kinect : Gestural User Interfaces add a Natural third dimension to HCI. In *Proceedings of the International Working Conference on Advanced Visual Interfaces - AVI '12*, pages 116–123. ACM New York, 2012.

[GLNM12] Kathrin Gerling, Ian Livingston, Lennart Nacke, and Regan Mandryk. Full-body motion-based game interaction for older adults. In *ACM annual conference on Human Factors in Computing Systems - CHI'12*, pages 1873–1882, Austin, Texas, USA, 2012. ACM Press.

[GMK+13] Andrej Grgurić, Miran Mošmondor, Mario Kušek, Carsten Stocklöw, and Dario Salvi. Introducing Gesture Interaction in the Ambient Assisted Living Platform universAAL. In *12th International Conference on Telecommunications - ConTEL 2013*, pages 215–222, Zagreb, 2013.

[GP12] Grzegorz Glonek and Maria Pietruszka. Natural User Interfaces (NUI): Review. *Journal of Applied Computer Science*, 20(2):27–45, 2012.

[HBC+96] Thomas T. Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. *Curricula Interaction, Human-Computer*. New York, 1996.

[Hed13] Bob Heddle. The new generation Kinect sensor is coming next year. http://blogs.msdn.com/b/kinectforwindows/archive/2013/05/23/the-new-generation-kinect-for-windows-sensor-is-coming-next-year.aspx, accessed in July 2013.

[Hei06] Rüdiger Heimgärtner. A Tool for Getting Cultural Differences in HCI. In Kikuo Asai, editor, *Human Computer Interaction: New Developments*, pages 343–368. 2006.

[Hev07] Alan R Hevner. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2):87–92, 2007.

[HMPR04] Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems. *MIS Quarterly*, 28(1):75–105, 2004.

[IKK12] Itauma Isong Itauma, Hasan Kivrak, and Hatice Kose. Gesture imitation using machine learning techniques. In *20th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, Mugla, April 2012. IEEE.

[KBR+07] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. Ambient Intelligence in Assisted Living : Enable Elderly People to Handle Future Interfaces. In Constantine Stephanidis, editor, *Universal Access in Human-Computer Interaction. Ambient Interaction: 4th International Conference on Universal Access in Human-Computer Interaction, UAHCI 2007 Held as Part of HCI International 2007 Beijing, China, July 22-27, 2007 Proceedings, Part II*, pages 103–112. Springer Berlin Heidelberg, 2007.

[KED+12] Felix Kistler, Birgit Endrass, Ionut Damian, Chi Tai Dang, and Elisabeth André. Natural interaction with culturally adaptive virtual characters. *Journal on Multimodal User Interfaces*, 6:39–47, January 2012.

[Ken04] Adan Kendon. *Gesture: Visible Action as Utterance*. Cambridge University Press, 2004.

[KK02] N Krahnstoever and S Kettebekov. A real-time framework for natural multimodal interaction with large screen displays. In *4th IEEE International Conference on Multimodal Interfaces (ICMI'02)*. IEEE, 2002.

[KP10] D W F Van Krevelen and R Poelman. A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality*, 9(2):1–20, 2010.

[LARC10] Paul Lukowicz, Oliver Amft, Daniel Roggen, and Jingyuan Cheng. On-Body Sensing : From Gesture-Based Input to Activity-driven Interaction. *Computer*, 43(10):92–96, 2010.

[Li12a] Y Li. *Hand gesture recognition using Kinect*. PhD thesis, University of Louisville, 2012.

[Li12b] Yi Li. Hand Gesture Recognition Using Kinect. In *IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)*, pages 196–199, Beijing, 2012. IEEE.

[LM213] Leap Motion: Product. https://www.leapmotion.com/product, accessed in July 2013.

[M2013] Kinect for Windows SDK. http://www.microsoft.com/en-us/kinectforwindows/develop/new.aspx, accessed in July 2013.

[MEVO12] Hossein Mokhtarzadeh, Amber Emmens, Peter Vee Sin Lee, and Denny Oetomo. Feasibility Evaluation of the Microsoft Kinect as a Low Cost Alternative to Motion Capture Systems. 2012.

[MMMM13] C McDonald, M McPherson, C McDougall, and D McGloin. HoloHands: games console interface for controlling holographic optical manipulation. *Journal of Optics*, 15(3), March 2013.

[Mor13] Leonel Morgado. Cultural awareness and personal customization of gestural commands using a shamanic interface. In *Proceedings of the 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2013)*, volume 00, Vigo, 2013. Elsevier B.V.

[MS213] Microsoft Kinect. https://www.microsoft.com/enus/kinectforwindows/, accessed in July 2013.

[NUI13] NUI. http://www.frankwatching.com/archive/2009/05/30/surface-interactieve-tafel-rijke-ervaringen-met-natural-user-interface/, accessed in July 2013.

[ope13] OpenNI. http://www.openni.org/, accessed in July 2013.

[Pau11] Bruno Campagnolo De Paula. Adaptando e desenvolvendo jogos para uso com o Microsoft Kinect. In *SBGames 2011*, pages 1–13, Salvador, 2011.

[PGGP13] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-Home Gesture Recognition Using Wireless Signals. 2013.

[Pin13] Pedro Pinto. Leap Motion Unboxing e Análise. http://pplware.sapo.pt/apple/leap-motion-unboxing-e-anlise/, accessed in July 2013.

[PS13] Benjamin Poppinga and Thomas Schlömer. wiigee. http://www.wiigee.org/, accessed in July 2013.

[Pu13] Qifan Pu. WiSee. http://wisee.cs.washington.edu/, accessed in July 2013.

[RA00] Jun Rekimoto and Yuji Ayatsuka. CyberCode : Designing Augmented Reality Environments with Visual Tags. In *Dare 2000*, pages 1–10, Elsinore, 2000. ACM Press.

[RBA08] Matthias Rehm, Nikolaus Bee, and Elisabeth André. Wave Like an Egyptian — Accelerometer Based Gesture Recognition for Culture Specific Interactions. In *HCI 2008 CULTURE, CREATIVITY, INTERACTION*, pages 13–22, 2008.

[RBN10] ML Rivera, NI Badler, and Aline Normoyle. Project PAALM: Phalangeal Angle Approximation through the Leap Motion Controller. Technical report, 2010.

[RL11] Matthias Rehm and Karin Leichtenstern. Gesture-based mobile training of intercultural behavior. *Multimedia Systems*, 18(1):33–51, July 2011.

[RM10] Darren Reed and Andrew Monk. Inclusive design: beyond capabilities towards context of use. *Universal Access in the Information Society*, 10(3):295–305, August 2010.

[RT13] Lorena A Rezende and Dalton M Tavares. OpenNI e Suas Aplicações. pages 371–378, 2013.

[SA09] Vipin Saxena and Deepak Arora. Performance evaluation for object oriented software systems. *ACM SIGSOFT Software Engineering Notes*, 34(2):1, February 2009.

[Sal13] Nintendo Co., Ltd. - Consolidated Sales Transition by Region. Technical report, accessed in July 2013.

[SLC11] Eduardo Souza Santos, Edgard A. Lamounier, and Alexandre Cardoso. Interaction in Augmented Reality Environments Using Kinect. In *XIII Symposium on Virtual Reality*, pages 112–121, Uberlandia, May 2011. IEEE.

[SLR+11] Evan a. Suma, Belinda Lange, Albert Skip Rizzo, David M. Krum, and Mark Bolas. FAAST: The Flexible Action and Articulated Skeleton Toolkit. In *2011 IEEE Virtual Reality Conference*, pages 247–248. Ieee, March 2011.

[SLR+13] Evan A. Suma, Belinda Lange, Skip Rizzo, David Krum, and Mark Bolas. FLEXIBLE ACTION AND ARTICULATED SKELETON TOOLKIT (FAAST). http://projects.ict.usc.edu/mxr/faast/, accessed in July 2013.

[SPHB08] Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll. Gesture Recognition with a Wii Controller. In *Second International Conference on Tangible and Embedded Interaction (TEI'08)*, pages 11–14, Boon, Germany, 2008.

[Sua10] Daniel Suarez. *Freedom*. Quercus Publishing Plc, London, 2010.

[Szi09] Gabor Sziebig. Achieving total immersion : Technology trends behind Augmented Reality - A survey. In *9th WSEAS International Conference on SIMULATION, MODELLING AND OPTIMIZATION*, pages 458–463, 2009.

[TL213] Thalmic Labs: Myo. https://www.thalmic.com/myo/, accessed in July 2013.

[Val08] Alessandro Valli. The design of natural interaction. *Multimedia Tools and Applications*, 38:295–305, January 2008.

[VRS11] Norman Villaroman, Dale Rowe, and Bret Swan. Teaching Natural User Interaction Using OpenNI and the Microsoft Kinect Sensor. In *SIGITE'11*, pages 227–231, New York, New York, USA, 2011. ACM Press.

[Wil09] Ross Williams. *Finger Tracking and Gesture Interfacing Using the Nintendo ® Wiimote*. PhD thesis, University DeLand, 2009.

[XB12] Elias Ximenes and Odemir Martinez Bruno. Reconhecimento de gestos em imagens de profundidade utilizando Kinect. In *WVC2012*, Góias, 2012.

[YD10] Ying Yin and Randall Davis. Toward Natural Interaction in the Real World : Real-time Gesture Recognition. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI 10)*. Association for Computing Machinery (ACM), 2010.

[ZZH13] Yang Zhang, Li Zhang, and Alamgir Hossain. Multimodal Intelligent Affect Detection with Kinect (Doctoral Consortium). In Ito, Jonker, Gini, and Shehory, editors, *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, pages 1461–1462, Saint Paul, Minnesota, 2013.