Popularization of Gesture Interfaces

João Carlos Andrioli Machado

University of Tampere
Department of Computer Sciences
Interactive Technology
M.Sc. thesis
Supervisor: Roope Raisamo
June 2010

University of Tampere
Department of Computer Sciences
Interactive Technology
João Carlos Andrioli Machado
M.Sc. thesis, 49 pages
June 2010

Human-Computer Interaction (HCI) is a field of research that studies ways in which humans and computers can interact. Gesture interfaces is a particular area of interested situated inside this field. It is about HCI techniques based on the interpretation the users' movement of the head or limbs, and it also relates to haptics.

This thesis examines the field's advances made until the present day and analyses how the people have benefited from it. The aim of the research is to find if gestural interaction has been propagating, its use increased, and identify ways of disseminating gesture interaction and HCI to the general population. A further concern is to improve the distribution of the benefits from HCI and gesture interfaces, reaching out also to a population that is often left disregarded – the developing nations.

This is done by reviewing previous work on gesture interfaces under HCI and Computer Vision. This provides the grounds for defining the ideal technology or technique which can facilitate the development of gesture interfaces that are accessible to ordinary users, and can be used in their routine. This definition helps identifying the technology which can best provide for massification of gesture interfaces: the Nintendo Wii Remote controller (Wiimote).

After selecting the Wiimote as the best device, it is examined in order to analyze what deficiencies and qualities it has. The technologies it carries are also looked into in detail in order to comprehend the possibilities of connectivity with other devices.

This technology is put to test and evaluated both from technical point of view (implementation learning curve, tools, flexibility), and user experience (usability, stress, efficiency). A series of recommendations follow on how to proceed with further implementation of tools and applications that assist users and current interaction models, and can help to change the way they interact with computers.

After the initial steps of constructing a test application, it was concluded that the Wiimote is not an appropriate device for this application to build on top of. The proposed approach and design cannot accomplish the goals of a gesture interface due to problems related to the dependency on a fundamental external component.

Although the Wiimote is clearly fit for many gesture interfaces, its use and applications are limited, it can only serve a definite number of scenarios.
Key words and terms: haptics, Nintendo Wii, gesture, computer vision.

Contents

1. Introduction 1
 1.1. Context and problem statement 3
 1.2. Research questions 4
2. Gestural interaction 7
3. Computer Vision 13
4. Popularization of gesture interfaces 18
5. Identifying a technology 23
6. Technology analysis 27
 6.1. Motion sensing 28
 6.2. Interaction models/methods/techniques 29
 6.3. Limitations 30
 6.4. Tests 30
  6.4.1. Wiimote for pointing and text input 30
  6.4.2. Technology applicability and interface feasibility 31
  6.4.3. Development case 33
7. Evaluation and Discussion 38
8. Summary 43

References 46

1. Introduction

It is essential to the progress of science that researchers use great amounts of resources to achieve certain capabilities or certain goals. Although it is often the case that only a fraction of the world may directly benefit from or participate in research, there are also fields where the outcome of research can be accessible to a much wider population.

One field in particular has good potential for such opportunities: research in Human-Computer Interaction (HCI). According to [ACM], HCI "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". A basic goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs. A long term goal of HCI is to design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

Computers and information science are dramatically increasing their presence in everyday life, so it stands to reason that improving human-computer interaction has become a central issue. The changes in how people use computers and what they are used for open new sets of possibilities for innovation in how the two parties interact. It does not only allow new ways of interaction: it requires them – indeed there is an increasing demand for easier and more rewarding interaction experiences [Mäkinen, 2007]. Whether it is with a desktop or laptop computer, a smartphone, gaming console, PDAs or a home entertainment system we interact with these machines frequently and everyday.

Popular research in HCI includes exploring innovations in gesture recognition, voice recognition and synthesis, touch-sensitive displays or haptics, multi-touch interfaces and force feedback. HCI combines with other fields of research to achieve certain more specific goals; such fields may be, for example, computer vision where the computer is able to identify (see) the user or his actions. The potential of this field is practically unimaginable since it can merge with innumerous other fields to allow revolutionizing interaction experiences. It is indeed an interesting interaction experience that certainly attracts attention of users.

HCI improvements are especially desirable in situations where devices (or hardware in general) are too limited to offer a satisfying experience. The mobile phone industry has been well aware of this issue for some years. Although they have been trying to address it, not many solutions have been presented so far; actions come most commonly in the form of user-interface design, resulting in less-than-desirable efficiency and incompatibility with desktop computers' paradigms. Most manufacturers kept trying to exploit the traditional interaction methods as much as possible, providing phones with extended keyboards for instance. A few solutions, however, really provide answers with the help of HCI – one example is the iPhone, innovating with a multi-touch screen. One area in particular with an enormous demand and potential for HCI improvements is entertainment. Gamers have been waiting decades for virtual-reality immersion, head-tracking, gaze-tracking, gesture recognition, not to mention other interaction techniques [Grossman et al., 2009]. Although it may not have such a considerable demand as the entertainment industry, health also has space for HCI improvements, especially in speech interfaces such as for the handicapped, robotic-surgery assistants and simulators.

Products with improved or special interaction implementations are already available to some extent. As mentioned before, the possibilities are practically endless for HCI improvements [Hemmert, 2009]. The given examples do not mean, however, that all average users would benefit from those products. In fact, until today the released products or published work under HCI have a very limited public. Even though the available products are out there, ordinary people do not want to or cannot purchase them (this is particularly evident in developing countries) – normally the widest population does not benefit from HCI advances. Examples of this are the work of Bernardes et al. [2009], whose solution has not been integrated in the official product *enJine*; Lee et al. [2007] whose tools are not commercially available as a toolset or individual products; and Lee et al. [2008a] also not commercially available (except for one part of the system).

Although not an exclusive problem of the field of HCI, it appears that there are difficulties in converting the outcomes of research into real-world solutions. Even when a research achieves practicable or reusable results, they're often not accessible to the widest population because of certain constraints (a requirement for a specific technology or device, or simply because the research is specifically aimed at a very strict domain, scenario or subject). It is notable that even though some products are inspired in HCI research, it normally takes a long time between the publication of a research and the release of a product based on academic work (Lee, 2008] stated that EA Sports proposed the release of a gaming system based on his inventions).

Considering the changes that the information society is going through on how to use computers and what to use them for, it stands to reason that new human-computer interaction experiences are a central concern. However, how are these new ways of interacting realized? There is a need for tools (i.e. assisting humans to train the computer) and applications (that make use of the specialized interaction and implement the experience). Most importantly, how can these tools and applications be made available to the widest population? We need to bring HCI innovations to new scenarios, where the users bring their limitations (instead of hardware), such as purchasing power. In an increasingly globalized, socially responsible and integrated world, it is also important to consider the less privileged audience: their limitations, use-cases, contexts and scenarios can introduce new needs and ideas that could not be devised if one were only to consider a developed and well-resourced public.

1.1. Context and problem statement

Most current computers are based on the traditional keyboard-mouse input methods, which has been reported to be hard and a cause of frustration to users [Mäkinen, 2007]. There is a plethora of guidance in the HCI field on how to make interactions easier and better; for instance Jacob Nielsen's usability heuristics [Nielsen et al., 2003]. Ultimately, however, no matter how well designed and created an application is, it will eventually face the limitations of the interface (interaction or input method). The solution to push these limitations is to introduce new ways for the users to interact with computers – either by providing new equipment and/or specialized software.

Much of the reviewed work focuses on gesture interfaces. In fact this area has been drawing much attention and interest due to the nature of the interaction that it enables. Wexelblat [1995] noted that "people are quite fluent at making multimodal presentations involving speech, gaze, sketching, and gesture". He explains that this is done fluidly in everyday conversation, moving between the modalities as appropriate or required; and defends that gesture is an especially important supplement in making ourselves understood. Kallio et al. [2006] also state that gesture interaction is a potential complementary modality; they reason that the number of different types of terminals brought to the market, with the latest technological advances, created a demand for new HCI methods – especially the gesture-based. It is easy to observe that people use gestures considerably in their every day lives; it's a very natural and intuitive way to communicate.

The fact is that gestures are important for nearly everyone. Some cultures use more gestures than others, but even when people from completely different and unrelated cultures meet, it's easy to notice that individuals resort to gestures to convey an idea and express themselves. In this way, we can see that gesture is a universal way to communicate – even when people don't understand each other's language, they can "talk with their hands". It is expected that usage of hand-gestures to interact with machines should be beneficial to the users [Bernardes et al., 2009; Kang and Ikeuchi, 1994b; Latoschik, 2001; and Shinoda, 2009].

Research on gesture interfaces has heavily contributed to the discipline's state of the art in HCI, but it is a fact that ordinary users are still deprived of most of the benefits from this modality. One could question why, after decades of research, computers still do not integrate any gesture-based features. After all, even with the very limited mouse it is possible to implement gesture interfaces – for instance the Internet browser Firefox has a plug-in software that provides a gesture recognition mechanism.

For some reason it seems that HCI fails to reach out to the average user; at least concerning gesture-based interfaces. One can speculate on the reasons for this: it may well be because gesture interactions usually require a certain specialized hardware that enables that interaction (gloves, hand-sensors, etc). In addition, many studies are focused on very complex issues such as gesture recognition and interpretation (when the computer interprets a gesture and translates it into a command or procedure), which is not very easy to implement successfully. Furthermore, the focus on the previously mentioned factors (the research focuses) is often so strong that it is not a concern to implement usable interfaces targeting average users. It is arguable that there are market and business reasons why companies or universities do not seek to take research results further, into the market: reasons range from lack of governmental support (e.g., budget, grants) to the infeasibility of commercializing a technology due to the common practice of university work to release work under licenses that are too restrictive for enterprises. In short, there is a gap between the HCI and the end-users; a gap from research to actual products.

Additionally, much of the research done on hand-gesture interaction takes specialized technology or hardware as a starting point [Baudel and Beaudouin-Lafon, 1993; Grossman et al., 2004], and this results in a drawback: it ultimately excludes the public from developing countries to benefit from that research. When this is not the case, it's likely that the research has a very strict application scope, or an interface design that is not usable and feasible in every-day situations.

If we carefully analyze human gestures, we will see that their role in communication is actually secondary – people do not use gestures to perform complex operations, neither do they use gestures to control and manipulate things [McNeill, 2009a] (which would require telekinetic abilities). One could argue that the more advanced, complex gesture-based interactions could be simply invalid because of the cognitive load they demand. The complexity here frequently creates a new language, when users must learn signs (gestures) that the computer is able to comprehend – an analogy to this is the need for adaptation of the members in an orchestra to a new maestro.

The most natural way to use gestures with the computer would be the simplest possible – either without gesture recognition at all, or with very minimal gesture recognition (for cue and contextualized actions). There are two main reasons for this. Firstly that is how it would best resemble the real-world usage of gestures: they are simple, secondary and used for contextualization or aid. Secondly, the vast majority of users do not trust computers enough to believe that it could actually "understand" what they want to do. Computers traditionally have very little semantic understanding of the users' tasks, so they are used to have complete control of the interaction where everything must be explicitly conveyed to the machine [McNeil, 2009b].

1.2. Research questions

In addition to HCI issues in applications that implement gesture-based interfaces, there is still considerable amount of work to cover in gesture interactions.

The main question in this thesis is how to fill the gap between HCI research on gesture interfaces and the usage of gesture-based applications, how to provide normal users with a real gesture experience. The first objective of this thesis is to search for possible ways fill the gap between HCI research and end-users; minding to provide also for users with limited resources. If the first objective is met, a secondary goal is to implement a relatively simple gesture interface, with the objective to evaluate a technology which has been found to be the most suitable. It's important to analyze how efficient and comfortable it is to use the interface in daily tasks. Another central issue is to test and verify if it is practicable to use a gesture interface heavily, and daily. An aspect of equal importance is ensuring accessibility for the widest population, so this thesis is concerned with a public that may not have an ideal purchase-power.

This is done by:

1. Defining how a technology can help disseminating and popularizing gesture interfaces and thus also the HCI field;
2. Studying how it can facilitate bridging the aforementioned gap. This will result in the definition of requirements and desirable qualities for technologies;
3. Identifying at least one technology that fulfils to the previous requirements;
4. Analyzing the selected technology in terms of implementation; evaluating the technology; describing what its strong qualities and its faults or problematic issues are;
5. Proposing design(s) for applications that could test the gesture interface.
6. Producing software that allows testing at least the technology's basic features. Evaluate development progress, and the user experience by conducting surveys;

In chapter 2 we look at the concepts, definitions, and issues related to gestural interaction. A review of previous work is initiated, introducing problems which might be faced during the development of the current work, and solutions that could be possibly re-used. In the following chapter a significantly large part of gesture research is reviewed – computer-vision based gesture interaction.

The third chapter will examine the popularization of gesture interfaces, looking into what kind of products have been release into market, and what technologies have been used. From these observations we define requirements for an optimal technology, which would aid in achieving the goals of this research.

In the fourth chapter the Nintendo Wii remote-controller technology is identified, and observed that it matches the previously defined criterion. Next, this technology is analyzed with a special attention on the technical level, and its positive and negative points are examined. A set of tests is then proposed, along with an attempt to build the application which would verify the characteristics of the technology, and its feasibility and potential as the solution for the gestural interfaces problem.

The development of this work is then discussed in a separate chapter where some conclusions are drawn as to why the use of the Nintendo Wii remote-controller has failed in the proposed design and application. In the subsequent chapter, a summary is presented with the results of the research, indicating the most important assessments and facts which can be extracted from this work.

2. Gestural interaction

Gesture interaction is a research topic that has been present for decades in the field of HCI. Its prototypes range from specialized applications like sign language interpretation to general gesture recognition [Nielsen et al., 2003]. Most of the research concerned with gesture interactions focus on the recognition and tracking of the human hand, hand motion (gesticulation), or interpretation of gestures [Baudel and Beaudouin-Lafon, 1993; Gunes et al. 2004; Kallio et al., 2006; Kang and Ikeuchi, 1994b; Kela et al., 2005; Latoschik, 2001; Lee, 2008a; Nickel and Stiefelhagen, 2003; Prekopcsák et al., 2008]. Most research is very specialized, scoping a determinate application area or domain [Grossman et al., 2004; Kang and Ikeuchi, 1994a; Lee et al., 2008b; Lee, 2009; Morency and Darrel, 2006; Morris et al., 2006; Shinoda, 2009; and Strachan et al., 2004].

The reasons why gestures are important are well explored: only 7 percent of our communication is verbal, the most expressive way through which emotions are displayed and messages conveyed is facial and body expressions [Gunes et al., 2004]. It is only natural and logical, as one aims to make computers more human-like, to take into gesture. It has been reported that “there is good reason to think that non-verbal behavior will play an important role in evoking social communicative attributions” [Gunes et al., 2004]; humans are more likely to consider computers human-like when they display appropriate non-verbal communication.

It is also stated that “understanding human emotions through nonverbal means is one of the necessary skills both for humans to interact effectively with each other and for computers to interact intelligently with human users” [Gunes et al., 2004]. Additionally, it has been affirmed that in noisy situations users might depend on more than one modality and resort to non-verbal alternatives such as gestures. Wickey [2007] also points out the fact that when speech is ambiguous, listeners rely on gestural cues.

Although their research focuses on multimodal interaction, Gunes et al. [2004] contribute on the gesture topic by doing a pragmatic and practical analysis the topic. There is a strong focus on the communication enabled by gesture:

> The essential nature of gestures in the communicative situation is demonstrated by the extreme rarity of ‘gestural errors’. That is, although spoken language is commonly quite disfluent, full of false starts, hesitations, and speech errors, gestures virtually never portray anything but the speaker’s communicative intention. (…) speakers may say *left* while meaning right and the gesture most likely will point right.

Gestures are important in our communication, but not all movements we make convey a message to a listener. Gestures have been studied and classified into categories, where the categories relate to the semantics of the gestures in a communication channel between speaker-listener. It is also stated that "the concept of gesture is loosely defined and depends on the context of the interaction. Gestures can be static, where the user assumes a certain pose or configuration, or dynamic, defined by movement [Gunes et al., 2004]".

It is clear that gesture is an essential modality of human interaction, permitting efficient communication which would not be possible otherwise. It comes to reason, however, that the application of gesture to informatics must be carefully placed because of the several issues connected with ambiguity and imprecision.

The topic of gesture interaction can be structured in a few different bodies of research; there is research on technical aspects of machinery and devices, on methods for identifying gestures, on gesture classification and identification models, and on applications of models, techniques and technology. There are two main approaches concerning hardware aspects, one implements gesture interaction via signal processing, another via computer vision. There are alternatives to these options, for instance algorithms that implement gesture recognition on mouse gestures, or other research that joins the two mentioned together – but these are not common.

Work by Kang and Ikeuchi [1994b] follows the former approach: it resembles signal processing in that it does not try to "see" the user – it rather analyses the data produced by the input hardware. Although they have been much cited for their work, having produced a fruitful research with positive results, they acknowledge that average users couldn't easily familiarize with their design. This conclusion supports the work of Prekopcsák et al. [2008], who recommend important design principles that should be followed in the design of gesture interfaces – the focus here is in simplicity. While their work focused on "everyday hand gesture" interfaces, Wickey and Alem [2007] worked on collaborative gestures in scenarios where users interacted remotely; their recommendations focus on issues that concern the remote part of the interaction rather than the gestural issues, specifically. Important points to note from these recommendations are:

- Gesture systems should support easy composition of gestures, in order to allow users to customize the interface [Wickey and Alem, 2007].
- Since gestures normally allow a higher degree of freedom of movement and positioning, they should be also ubiquitous. Although there are several technological barriers to cross before this can be achieved, it is still an important issue to consider [Prekopcsák et al., 2008].
- Unobtrusiveness (or inconspicuity) is a highly desirable quality because it reduces the cognitive load on the users [Prekopcsák et al., 2008].

- The system should integrate control functions, in order to allow the user to define how the interface is controlled (without recurring to other means of interaction such as keyboard/mouse) [Wickey and Alem, 2007].
- The interface should be able to adapt itself to its users, not the other way around. Users have preferences such as being right- or left-handed, and also cultural differences which must be a problem for the system. Complex methods such as machine learning techniques might be necessary in order to deliver the required adaptability [Prekopcsák et al., 2008].
- The system should be simple enough to be set up in a few minutes. It is important to identify and disregard misuse, and provide feedback about successful gesture recognition.

The present work endorses the propositions of Prekopcsák et al. [2008] and Wickey and Alem [2007], mostly because it comes to meet the discipline of Usability, which is essential when it comes to user interaction. Although their research focuses on hand gesture, the design principles proposed can also be applied to other applications in the field as well.

Such interfaces can be represented by Figure 1, which depicts a user interacting in a natural way – he realizes simple swinging movements, without attached or wear-on hardware, in a non-restricted area.

Figure 1 Gesture interaction - sensing capabilities

Gestural interfaces can be implemented with a variety of devices and equipments that use more than one technology. An example application is the *Colabdraw* by Morris et al. [2006], where the table desktop is a multi-user touch-sensitive interface that also uses information from the users' chairs. Their work focuses on the analysis of cooperative gestures (not on the technologies), exploring the motivations for the user of cooperative gesture techniques. They expand gestural interactions by studying the aspects of cooperation in the gestures with multiple users. The work of Morris et al. [2006] contributes to the field by discussing motivating scenarios for the use of cooperative gesturing, identifying design issues relevant to these interfaces, and presenting a preliminary design framework.

Because gesture identification is usually conceived while having a limited set of gesture cultures or applications in mind, this is usually an application area that can be implemented in practically any software-engineering approach – some choose to limit the interaction with defined sets of gestures, others use AI to recognize gestures against a neural network database.

For example, Latoschik [2001] presents a framework for gesture detection and analysis that permits modelling multimodal interaction that includes gestures – the goal is to allow complex networks of movements, which form gestures (a concept similar to neural networks) – a similar proposition to the work of Gunes et al. [2004]. This work is an analogy to artificial intelligence systems that are generic enough to be used in varied situations. The focus of his research is on a modular gesture detection and evaluation system, applicable to virtual-reality environments. They do not aim to provide an interaction framework or set of applications that can be applied to ordinary computers and day-by-day usage.

Contrary to Latoschik [2001], Kang and Ikeuchi [1994a] propose a focused, very specialized solution to program robots with the use of gesture recognition. The technique is about segmenting the grasping tasks of a human into sequences of motions, structuring and formatting this information to program robot movements. They provide valuable knowledge on how to determine breakpoints in human hand motion while performing a sequence of tasks. Although Ikeuchi has an extensive research background, his focus is in automation; hence not much direct effort is placed on creating a reusable asset, such as a gesture interface framework.

One of the most remarkable works is the *Charade*, by Baudel and Beaudouin-Lafon [1993] – a system for gesture recognition that is actually made to be used by ordinary users. They designed a solution for “remote control of objects using ‘free-hand gestures’”. The system consists of a generally simple piece of extra hardware, the *DataGlove* (a glove equipped with sensors all along from the wrist to fingertips) and software that interprets the gestures, translating them into commands. Even though their work succeeded in creating a good application for average users, they conclude that it is not viable for day-to-day usage because of the additional piece of hardware. Their work is valid as the interaction technique can still be implemented with other “off-the-shelf” hardware and software.

In all the studies reviewed one particular difficulty was common: training the user. One research in particular, by Kallio et al. [2006], proposes solutions for visualizing hand gestures, which comes to aid users in familiarizing themselves with the system. This could potentially improve the “training stage” up to a point when the very training would become almost unnecessary. They report positive results by separating acceleration components, eliminating tilt effect during gestures, and integrating projected signals. It contributes to the field because it proves that “visualization clearly provides information about the performed gesture, and it can be utilized in providing essential feedback and guidance to the user" [Kallio et al., 2006]. Most importantly this material provides very good guidance on how to deliver user-guidance in gesture training.

When corporations have to go through software or hardware upgrades, they carefully plan the operations and investigate how the changes will affect legacy systems – drastically changing user programs and interfaces is known to be a bad move: because people are normally resistant to changes to what they already know how to use. In the same way, gesture interfaces must be implemented with caution, they shouldn't be implemented just for the sake of technology even though it may be tempting (in terms of implementation effort (cost) or business reasons (profit)). Similarly, when introducing gesture interfaces for ordinary or daily tasks, it must be carefully idealized. Users may not be prepared to use the full power of gestural interaction, so it should take place in the simplest way.

Even though some of the previous work offer very good classification of gestures and a provide basis for interpretation of gestures [Gunes et al., 2004; Strachan et al., 2004], in this thesis there is no intent to examine gesture recognition. The current view is that at this point in time, users are not habituated or comfortable with the computer being able to "understand" what the user means by a wave of arm. Generally speaking the average user doesn't trust the computer and is frequently afraid of the machine. Here, it's considered that a certain degree of evolution is yet needed before users can be offered experiences where the computer can "understand" them. This reasoning comes to justify the decision to analyze user interaction with a raw, rustic and simple gesture control experience. In other words, the interaction models to be focused on are those where the user doesn't transmit encoded commands via gestures, but have direct control and manipulation of the user interface.

Gestures have been proposed and put into use in many different systems and interfaces, including interfaces for people with physical disabilities. The benefit of eliminating the need for pointing can save time and effort; but performing for long periods of time has been discussed to be stressing [Cabral et al., 1995].

The signal-processing-based gesture interfaces have been criticized for not being comfortable and sometimes not efficient enough [Baudel and Beaudouin-Lafon, 1993]. Some of the hardware like the *DataGlove*, simply does not provide enough information about the user's movements. Furthermore, designs of wear-on hardware pieces frequently have the inconvenience of being wired to a computer (which restricts the user's mobility). Perhaps an ideal device would be one that could provide maximum gesture information while being portable and wireless.

In short, the literature survey reveals significant work in many areas of HCI (concerning gesture interactions) that one can build on when designing interfaces. Research normally focuses on studying the interaction technology, technique, or method; the design and construction of a framework or platform for gesture interfaces is normally not the aim of HCI research. The result is that there is no standard for developing gesture interfaces, and each research may come up with solutions or approaches that are similar – but there is too little synergy. The contribution of the present work to the field comes as it strives to combine previous research (as guidelines) in order to define a technology that can best provide for gesture interfaces.

3. Computer Vision

Computer vision is the science and technology of machines that see. As a scientific discipline, computer vision is concerned with the theory for building artificial systems that obtain information from images. The image data can take many forms, such as a video sequence, views from multiple cameras, or multi-dimensional data from a medical scanner.

As a technological discipline, it seeks to apply the theories and models of the field to the construction of computer vision systems. Many studies have applied computer vision to the field of gestural interfaces. This approach to the problem of gestural interfaces has several advantages; but suffers from many drawbacks.

Wikipedia summarizes:

> The field of computer vision can be characterized as immature and diverse. Even though earlier work exists, it was not until the late 1970s that a more focused study of the field started when computers could manage the processing of large data sets such as images. However, these studies usually originated from various other fields, and consequently there is no standard formulation of "the computer vision problem." Also, and to an even larger extent, there is no standard formulation of how computer vision problems should be solved. Instead, there exists an abundance of methods for solving various well-defined computer vision tasks, where the methods often are very task specific and seldom can be generalized over a wide range of applications.

The application of computer vision to gestural interfaces allows the users a high degree of mobility, movement, flexibility, and freedom – since the user input is normally captured by cameras or other means that do not require the user to have physical connection with the machines (nor wear special gear). It allows for ubiquitous computing and hands free interaction. Applications normally belong to virtual reality and augmented reality areas [Nielsen et al., 2003].

The many possible ways to “see” the human hand are discussed by Wang et al. [2007]. They criticize the drawbacks of attaching hardware to the hands and defend the usage of cameras and imaging algorithms to detect gestures – allowing free-hands interaction. In fact many researchers have a similar reasoning [Kang and Ikeuchi, 1994a, 1994b; Lenman et al., 2002; Morency and Darrel, 2006; Morris et al., 2006; and Wickey and Alem, 2007]).

The significant amount of research on the field of gesture under computer vision (especially on extracting facial motion, interpreting human activity, and recognizing hand/arm gestures) considers both static-position (pose) and dynamic (spontaneous movement) gestures. The concept of gesture, however, is loosely defined – it depends on the context of each particular interaction [Wickey and Alem, 2007]. It is interesting to note that many studies have been conducted without mentioning these important differences and issues that concern the topic.

Wickey and Alem [2007] outline the science of recognizing and interpreting gesture in the following components:

- Sense position, configuration and movement through computer vision techniques;
- Preprocess (image normalization, enhancement, or transformation);
- Gesture Modelling and Representation (transformation of the input into appropriate representation);
- Feature extraction and Gesture analysis (statistical properties or estimated body parameters);
- Gesture Recognition and Classification (template matching [HMMs or Bayesian networks])

There are many different possibilities for approaches to computer-vision-based gesture interfaces. As an overview of these approaches and techniques, the following is presented:

Features representation techniques analyse trajectory; motion, colour, intensity, moment (physics movement), borders, silhouettes, contours; or by parametric eigenspace representation. *Feature detection and localization techniques* use various techniques such as segmentation, filtering, edge detection, morphological skeletinization; and motion analysis.

Gesture Recognition Techniques can be classified according to three major approaches: model, appearance, and motion based [Wickey and Alem, 2007]. The first focuses on recovering three-dimensional model parameters of articulated body parts. Appearance based approaches use two-dimensional information such as gray scale images or body silhouettes and edges. The latter attempts to recognize the gesture directly from the motion without any structural information about the physical body [Wickey and Alem, 2007]. A typical way to address the temporal resolution is to use Hidden Markov Models.

Static gesture or *pose recognition* are normally solved by implementing template matching, geometric feature classification, neural networks, or other standard pattern recognition techniques such as parametric eigenspace to classify pose [Wickey and Alem, [2007]. *Dynamic gesture recognition* requires an intelligent and robust consideration of temporal events, much like a video analysis and pattern tracking. The accomplishment of this is normally achieved via implementation of techniques such as time-compressing templates, dynamic time-warping, and Hidden Markov Models and Bayesian Networks [Wickey and Alem, 2007].

Lenman et al. [2002] study if “marking menus” can be used to develop autonomous command sets for gestural interaction. They offer a very powerful recognition system that can identify hands even in the most adverse scenes. One limitation of this approach is, however, that the interaction is limited to what the system offers in terms of commands – there is a set of commands that can be activated by certain hand poses. The hand is used as a wireless remote controller; the interaction is mapped to a very limited “language”. Although this design has a very limited flexibility, it is highly usable in the day-by-day. One could easily control a series of devices around the house, for instance. Its usage also applies to Skype (although this hasn’t been mentioned by the author), by offering users the means to control various features of the computer while a video-call is held.

Nickel and Stiefelhagen [2003] address the problem of recognizing 3D-tracking based pointing gestures, with the aid of computer vision: a camera constantly films the user, combining images from head orientation, arms and hands. They identify the very root problem in this type of interaction system (computer vision based gesture interfaces) – detection of the occurrence of the gesture in a natural arm movement; and the estimation of pointing direction. They make use of Hidden Markov Models, trained to detect the pointing gesture in a 3D input stream. Their contribution is a robust approach to real-time 3D tracking of head and hand using color and range information. The reported rate of successful identification of pointing targets was above 90%. The criticism to this work, however, is that there is no clear plan on integrating this system on ordinary applications, or producing any sort of software that can be used by ordinary users. One potential scenario for their solution is during video calls with users that have video cameras, in applications like Skype.

Cabral et al. [1995] and his group have introduced a simple but robust 2D computer vision based gesture recognition system. They analyze several issues related to computer vision based systems and propose a set of principles for good gesture interfaces. This work is a development from previous research, and is aware of the limitations of gesture interfaces; but these principles contribute to the field because they impose extra effort on the design part of applications development – they are ergonomic guidelines which orient user-related interests. Nowadays the technological aspects of most solutions are just a fraction of the problem to the so called “alternative” interfaces (or new interaction techniques); the development of good usability is what will clearly define the acceptance of new projects.

Work by Naik et al. [2006] contributes to computer vision by applying it to gesture interfaces. This is done by creating a system that recognizes gestures continuously (like a stream-video analysis tool). They also focus in virtual reality, but their research is interesting because of the approach on the detection of the gestures – they claim to achieve a 94% recognition rate by using state automata. One factor of success is that they limit the users' range of gestures to a predefined set of "commands"; which is not ideal, but is what actually enables such efficiency. The insight of this research is the usage of an "avatar" (information visualization) to represent the user in the virtual environment: this avatar reproduces the captured movements, showing the user what the computer is interpreting from the gestures. There is great meaning in the use of this "avatar", because the effects of this approach are that the user can modify and correct his movements in order to make the computer correctly recognize a gesture – in other words: the duty of transmitting a correct message is shared with the user (even if that's not the aim) because the user ends up changing his natural movements into something that he thinks is more adapted to the computer.

Wilson and Oliver [2003] present a research backed by Microsoft where they implement a stereo-camera system that allows gesture control over windows. They propose the user of perceptual user interfaces as a complementary to mouse-keyboard interfaces. They justify this by the various problems that computer-vision systems face: computation burden that is added, lack of robustness outside the lab, and unreasonable calibration demands. Their work innovates because it presents the stereo camera element with very fast vision algorithm that successfully recognizes hand gestures. Their work is also important because it concludes, through their experiments, that average and ordinary users are ready for new ways to interact with the computer, and most importantly are eager to do that. Their research is relevant in the field because "unlike other computer vision algorithms, theirs does not rely on fragile appearance models such as skin color of hand image classification which are prone to break when environment conditions change".

The reviewed literature shows that many have tried to face the challenge of recognizing gestures through computer vision; and although some claim successful research results, this approach is still a long way from being truly usable. Additionally, none of the authors mention the scene and usage conditions that could limit this model of interaction – it restricts users' mobility by requiring them to remain in the camera's area of capture.

There is no information given if the recognition algorithms were tested in different ambiences, such as sunny outdoors or much darkened rooms. The usage of computer vision places the additional and important problem of automatic real-time detection and tracking of the hand component [Wilson and Oliver, 2003], not only recognizing the gestures. As discussed by Wilson and Oliver, these perceptual interfaces are quite fragile because "they are often based on techniques sensitive to unique environmental circumstances (e.g., colour models that highly depend on the lighting conditions), rely on the use of multiple CPUs or specialized hardware, are usually installed and maintained in very limited quantities, and require laborious calibration".

There is a potential advantage, however: many computer sets nowadays are sold with a web-cam, which could suggest that most users already have the equipment needed for computer-vision based gesture interfaces. Internet video-calls also helped disseminate usage of web-cams – if there are so many potential users to computer-vision based gesture interfaces, the gap to be filled is in the distribution of software that uses that hardware.

In fact, this is an area of focus to be developed: among the reviewed literature there is no mention of focus on user hardware – while the scientists are pursuing solutions to the core computer-vision problems, a possibly potential opportunity is being neglected. As previously mentioned, several research solutions can achieve reasonable levels of success in recognition of gestures. A clear action needs to be taken in order to give these solutions field-approval – the only actual means to validate and verify the academic work is to test it in the real-world scenarios. It is possible to provide computer-vision based gesture interfaces to the general population if ordinary hardware is supported.

The benefits to the field would be valuable because new contributions from open-source communities (or third parties) might help to apply research to new ways of thinking: a possible solution to the current movement and location restrictions could be solved by an initiative to combine mobile phone cameras with web-camera in a fashion as to replicate an academic work [Wilson and Oliver, 2003].

As quoted in the beginning of this chapter, computer-vision is still a very wide area of research, lacking a standard formulation of “the computer vision problem”. The result is an abundance of task-specific methods to solve well-defined computer-vision tasks; leaving users orphaned of mature solutions that can be generalized over various applications.

4. Popularization of gesture interfaces

It is clear that people are eager to use new and interesting ways of interacting with machines – the iPhone's success is the latest factual proof of this. It must not be forgotten, however, that the interaction needs to be efficient. This is but one requirement for the new interaction techniques – one simply will not trade effectiveness and efficiency for *coolness*, nor robustness for *sexiness*.

Gesture interfaces are not new in computer science, but it has not been a real success in terms of production and consumer-level goods. The true popularization of gesture interfaces only started recently, in the last few years a vast population of consumers started to seek devices that offered gesture features. This is clearly visible today as more and more customers speak out their eagerness to try the latest products by Apply, Nokia, and others – several of their devices gained fame for the interfaces that integrated multi-touch screen, accelerometers, etc. These devices allowed the creation of applications that can transform them in *fake Star Wars light sabers*[1], applications that can make the telephone respond to gestures (shakes, twists), or other interesting gesture related interfaces[2] such as the iPhone maze game.

One considerable step in the popularization of gesture-based interfaces was the release of the Nintendo Wii video-game console; which remains an unchallenged market leader. Sony has been reported to be preparing the release a new device to compete with Nintendo, a launch that is just 4 years late.

The Sony Playstation 3 will release during 2010 a new controller technology that is in many ways similar to the Nintendo Wiimote. The Sony Move, however, will be integrated with Sony's EyeToy, a video camera[3]. This union will allow the console to extract more information about the user movements, such as dislocation in space and the direction in which the user is moving – something that is not possible in the Nintendo counterpart.

Microsoft has also felt the drop in sales as the Nintendo Wii became the top seller of video-game consoles. It is now working on a project called *Natal*, a new technology of controllers for the Xbox video-game console. It has already been introduced to the public at the E3 gaming conference, and Microsoft's bold marketing promotes the device as one of the most revolutionizing releases in many years:

> Introducing Project Natal – a revolutionary new way to play: no controller required. See a ball? Kick it, hit it, trap it or catch it. If you know how to

[1] Lightsaber App For Nokia N95 by Rcadden – http://www.youtube.com/watch?v=YwksUo6PKH0

[2] Nokia N95 apps Rotateme, Shakeme, Garmin by kolias89 – http://www.youtube.com/watch?v=s2Ge6M35zog

[3] Playstation Move – The most immersive gaming experience possible http://us.playstation.com/ps3/playstation-move/

> move your hands, shake your hips or speak you and your friends can jump into the fun -- the only experience needed is life experience [Microsoft, 2010].

This *controller-less* technology has been reported to be using a structured-light 3D scanner[4]. The skeletal mapping technology is capable of simultaneously tracking up to four users for motion analysis, with a feature extraction of 48 skeletal points on a human body at a frame rate of 30 hertz. The absence of a controller might become a difficulty factor to the adaptation since users are used to have controllers as support tools for their own orientation.

All these devices and technologies have great potential to provide gesture-based interfaces for a large amount of people. Some of them are yet to be tried and verified after their launch. Even the technologies that have already been deployed must comply with a determinate set of requirements (which might be still undefined) that will ensure the best potential for a global and wide provision for gesture interfaces.

The present work proposes to search for, define requirements for, investigate, and evaluate a technology that can help to propagate and disseminate the use of gesture interfaces. In other words, how to better popularize the use of gestures with computers. Although there are many applications and solutions available, most research failed to reach out to the great public – apart from the "political" and "business" reasons for this, several technical problems have been criticized previously.

The reviewed literature divulges that all the technological options have flaws. Some researches successfully apply all the advantages of a technology to a particular application; but the applications permitted by the technology are limited. The question remains then: what characteristics would be desired in a technology so that it can best support and provide for gesture interfaces?

Most dissemination of gesture interfaces has been taking place in the area of portable devices, such as telephones and laptop computers. The desktop computer and other home devices (i.e., TV set-top boxes) continue to focus mainstream interfaces on buttons.

One might argue that this development is happening in such way because the portable devices simply do not offer any better possibility; hence, the gesture interface gains space. It could be assumed out of this argument that the keyboard/mouse combination is already the best possible offer in HCI interfaces, but this would have to be proved while conducting studies to compare usable, robust, and properly integrated gesture interfaces with the traditional methods – which has yet to be done. While this

---

[4] PrimeSense Supplies 3-D-Sensing Technology to "Project Natal" for Xbox 360- http://www.microsoft.com/Presspass/press/2010/mar10/03-31PrimeSensePR.mspx?rss_fdn=Press%20Releases

research remains to be developed, we may explore another alternative: the possibility that gesture interfaces have not expanded because of inappropriate technology and applications.

Although several new handset devices (iPad, iPhone, Nokia N900) offer possibilities to detect direct user input and some detect multiple points of contact, the gesture interface is still missing. There are many reasons why these devices will not help on objective of expanding gesture interfaces accessibility; but perhaps the most obvious one is the price. The cheapest of these machines can be found for a minimum of 350€ which is by far out of range for most of Latin-American and African population.

A quick survey on a few developing countries will reveal that after taxation most of the handheld devices have their price so high that customers can compare them with prices of ordinary desktop computers.

These facts lead to a clear conclusion, that the technology capable to maximize user adoption and allow for gesture interfaces needs to comply with a minimal set of restrictions. Clearly, price is one of these restrictions, but it is not yet defined what the other restrictions are. In light of this conclusion, it stands to reason that a list of requirements be elaborated, and this will be discussed next.

Although there are many technologies available that permit gesture interaction, only a handful have qualities which allow them to be used in the amplest variety of situations, for example because it is specially designed for specific tasks such as surgery assistants or robot training. Producing a "generic technology" is difficult because at the time of design, one cannot envision all the use-cases which would need to be covered. Most often these technologies carry too many faults for not enough capabilities.

While some are unfit for the price, a few are accessible at the reach of ordinary people's purchasing power. This is an important issue because providing gesture interfaces "to the people" means that any required hardware and software must fall within the price range of other standard computer peripherals like mice, webcams, and game controllers.

Some of the references cited earlier include experiments where real users tested a system; and most of them were conducted in controlled environments: the laboratory. For the gesture interfaces to be widely adopted they must perform robustly outside of the laboratory. They must be tolerant and resistant to adverse environmental conditions and also to different users' personalities and habits.

If the interface is targeting the masses, it must be easy and simple enough to plug to a generic computer and "just work" (plug & play concept), because average users cannot perform complex, technical IT configuration or set-up tasks – unreasonable calibration demands are not acceptable. In other words, it must be a technology that relies on "standard hardware" implementations. Relying on common hardware allows the integration with most operating systems and machinery; allowing older computers to be used too.

Another important requirement is to be computationally inexpensive – this allows for the widest population to have access to the technology (because slower computers are suitable), permits developing nations to benefit from it as well. If the technology implements complex or computationally-expensive algorithms, it should be based on a device that can process that extra itself.

These interfaces should not rely on intrusive or cumbersome equipment such as gloves, headsets or close-talk microphones – it has been extensively discussed that this would considerably reduce the usability and user-friendliness of the systems [Baudel and Beaudouin-Lafon, 1993; Nielsen et al., 2003]. Instead, the technology should resemble simple and handy objects that users are familiar with; or even toys if possible – simplicity and comfort play an important role in the usability of a gesture interface.

From all the reviewed literature, there isn't a single piece where users wouldn't have the need to point. In fact, very few interfaces do not count on the aid of pointing – even amongst handheld devices, tapping or pointing are important for the users. Pointing is still necessary due to (among many other reasons) how computers have been used for the past decades; it became natural and comfortable, throughout various domains, to use pointing – it is now part of the very nature of the human-computer experience. It is only reasonable that gesture interfaces should also permit pointing – it must be precise, efficient, and similar to pointing with the mouse. It is a crucial topic because users depend on it (and it often is the only way) to control the interaction. It is a sensitive issue because it must be precise and accurate – there is not much tolerance for errors; pointing reliability and efficiency must be maximized.

As Nielsen [2010] evangelizes, giving feedback is very important and extremely valuable, therefore providing ways to output information to the user is an important benefit. Because the technology will never be aware of the application to which it's serving, it should expose its output capabilities in a clear way.

Giving feedback is not the only usability requirement, however – it is also important to allow users the maximum mobility. Using this interface shouldn't prevent users from moving in a room or limit their workspace to a strict area. Although computers are mostly used on a desktop, the freedom of movement and liberty of mobility are advantages that certainly make up for a better interface.

One more important plus is the usage of joint modalities. If the device implements a technology that benefits from different input/output modalities, the advantages increase to another level. Combining for example gestures with haptic interfaces, voice, sound, and other methods is extremely valuable because their combination achieves a synergy that one modality alone cannot.

Finally, in order to be a technology that really allows reaching out to the masses, it must (apart from being cheap) be available in virtually any country.

The support for gesture can be implemented mainly in two different approaches: based on computer vision (cameras and image analysis) or signal processing (sensors). Ideally, the most flexible choice would be the latter, since the former is more restrictive in terms of user dislocation and ambience conditions. The several drawbacks and advantages of computer-vision based systems were discussed previously. Signal processing alternatives usually suffer from cable length limitations or other distance restrictions. The best solution would be then to combine the most important and desirable advantages of computer-vision systems (freedom, ubiquity) with gesture interfaces based on sensors (accuracy, precision). A potential synergy between the two approaches is that one could help to provide where the other lacks and vice versa.

While any given technology can suffice for gesture interfaces without necessarily fulfilling all these requirements, it is important to highlight that meeting them will mean that the technology is closer to reaching the widest span in application domains. Thus, potentially, could help to change how users interact with computers nowadays and influence the new generations.

5. Identifying a technology

Up to now we have explored the motivation to span the usage of gesture interactions to a wide public (and how beneficial to the field that would be); gesture interface studies have been analyzed; options for the realization and to bringing gesture interfaces to the masses were discussed. After defining the requirements for a technology which can best suit this purpose, we will discuss how a technology has been identified.

Previous work has also contributed to the review of devices; and as reported by Wong et al. [2008], the conclusion is that no other device is as accessible as the Wiimote (see Figure 2).

The Nintendo Wii gaming console has been drawing constant attention over a considerable period of time. It innovates in HCI with a gesture-based controller that changed the way users interact and play games.

There is no question about Nintendo's penetration in the market – it has been the best selling console for the past year, and holds the best selling games position of January 2009 [Lee, 2009]. Shops can't keep it in stock, but some say that this may be a market strategy to drive up demand by withholding supply [MacDonald, 2009]. Wii games are not just the best selling, but are so by a vast margin. No other technology offers a comparable level of user adoption and pervasiveness. It is arguable that gesture interaction has long been studied and researched in computer science, but failed to reach the masses until the launch of the Nintendo Wii.

Figure 2 The Nintendo Wii Remote connected to a Nunchuck

With the popularization of the Wii, it became evident that people are ready and eager to try new ways of interacting with machines. The success of the Wii cannot be attributed to the single fact that it implements gesture/haptics interaction, however, it is a combination of many factors (haptics interaction, gesture interaction, user-friendly design, etc). The inspiration from this easy-to-use gesture interface has resulted in hundreds of new applications being produced and readily available online for the most curious haptics-enthusiasts.

The Wii users interact with it through a controller called "Wiimote". It consists of a hand-held remote controller that looks like a candy bar. It is simple to use because there isn't much of a learning curve to it – as a tool it's mainly used to point, and for controlling the interface. There are some applications of gesture recognition to control the interaction, but the vast majority of applications implement simple, direct-control interpretation of signal processing. In other words, the user's movements have immediate and direct influence and results on the interface. It is a simple handle to the virtual world, because of a simple and comfortable design, the controller doesn't feel cumbersome; there are no cables; and it's not intrusive.

This is the one single equipment which combines the two most successful methods of implementing gesture interfaces – signal processing from the accelerometer (see Figure 3), and computer-vision with the infra-red camera [Wiire.org]. The computer-vision approach permits precise pointing functionality, which can also be used for other types of interaction, not only pointing. The infra-red camera's "vision" can be directly accessed and the interpretation of its viewpoints can be programmed at will.

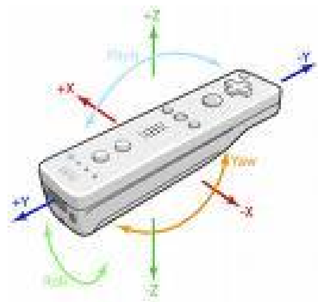

Figure 3 Wiimote and its motion sensing capabilities

The Wiimote carries the implementation of a HID[5] device, which means it is a standard device implementation that is capable of *Plug'n'Play* over Bluetooth radio. In other words, it is possible to connect the Wiimote to any computer that has a Bluetooth radio dongle, and by using HID interface one can retrieve information from the Wiimote's sensors, the camera, and also send data to the speakers and vibro-actuator.

The embodiment of the controller is very resistant, and the latest releases have come with a protective rubber which makes it a very resistant device. But not only the physical aspects are robust: the Wiimote can perform well in almost any situation – at least what concerns the gesture part. There may be situations (like sunny outdoors) where the computer-vision implementation may suffer due to the strong light and reflections of infra-red light from different sources. In general, however, it is a device that sufficiently isolates external noise and interference.

Because of the Bluetooth radio the Wiimote allows users a new level of freedom of movement and independence from the machine world.

---

[5] Described in http://www.usb.org/developers/devclass_docs/HID1_11.pdf/

The Wiimote is equipped with two ways of feedback: it contains a device for basic audio output, and a "rumble" vibro-actuator. There are not any defined guidelines for designing applications that make use of the Wiimote, but one can try at best to mimic the design approaches that have been used in the Wii console and its games; for they are very successful (although there are some games which really lack usability analysis). For the simplicity of use (HID, *Plug'n'Play*), and the satisfying range of feedback options, the Wiimote design achieves a significant level of facilitation for usability.

The Wiimote embarks much computation – it provides computed information about accelerometer data via reports [Wiire.org]. The computation here translates the accelerometers raw data into predefined-range values that are reported to the connected party through the reports-style protocol [Wiili.org, 2009a]. It does the same for the infra-red camera's data, interpreting the visual signals and translating them into coordinates that are communicated in the reports. In a similar way it allows transfer of raw sound data to the speaker system for audio-output on the controller. The vibro-actuator can also be activated with the same report-like feature (see figure 4 [D]).

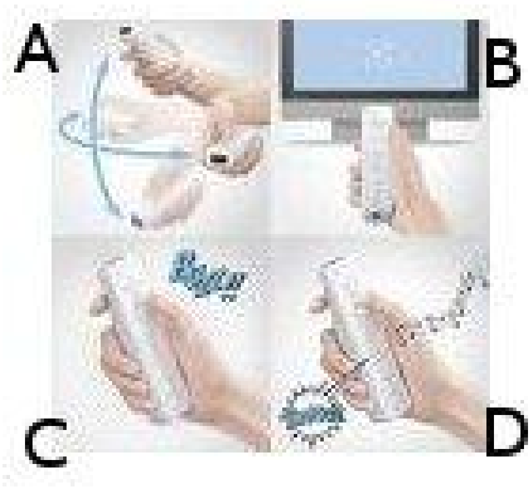

Figure 4 The Nintento Wii Remote controller

This description reveals that the Wiimote device is actually reasonably simple – it simply reads data from different sensors, calculates and translates signals into definite structures or values, and sends these "pre-cooked" pieces of information to the connected host. Interpreting this information and making use of it in a gesture interface is then a matter of choosing the appropriate technique. The result is that using a Wiimote with a standard computer is computationally inexpensive, because it doesn't require processing the sensors' data.

The Wiimote has an extension port where it is possible to connect several extensions: the most popular of them being the nunchuck (present in every Wii system). The additional port provides an interface to different extra gadgets that can allow for more complex interactions. The nunchuck is the first attachment revealed for the

Wiimote in 2005; it connects via a long cord, and resembles a nunchaku. It features an analog stick and two trigger buttons, and it also provides accelerometer data for motion-sensing, but it has no rumble features. The second most popular attachment to the Wiimote is the “Guitar Hero Guitar”, a toy that resembles a guitar, containing a few buttons as an analogy to the guitar’s notes. Several other extensions are available, such as the Nintendo “Classic Controller” (which is actually a one-piece controller with components of the Wiimote and Nunchuck merged together); the latest extra-motion sensor attachment, etc [Wiili.org, 2009a, Wiire.org].

These attachments are the confirmation that the Wiimote has been designed to be scalable and upgradeable, supporting future inventions and innovations. It surely will face limitations due some technical restrictions of its hardware; but as of today, it hasn’t even been extensively explored in gesture interactions and new applications for this technology are appearing constantly.

This technology is the one which best fulfills the requirements cited earlier. It is the single technology with better potential and most possibilities. It is a very cheap equipment, considering all the technological components it carries – normally at about 45€ in Europe. Other previous studies [Yim et al., 2008] mention that tracking systems are typically based on expensive specialized equipment (which motivated the creation of low-cost solutions); hence the Wiimote is an alternative of strong competitivity. Yim et al. [2008] cite some options that range from US$200 up to US$28.000.

For the reasons discussed here, the Nintendo Wiimote has been selected as the subject of deeper study. We will now analyze what applications it can support, how it can be extended, and how easy it is to produce applications that make use of it.

6. Technology analysis

The Wiimote details have been shortly discussed, and its components shortly introduced. Here we take a more technical look into how it works, and analyse how the technical details could affect an interface. As mentioned earlier, this is an implementation of a standard HID device [Wiili.org, 2009a, Wiire.org], communicating to the console over radio frequency. The Wiimote has the ability to sense acceleration in 6DOF via an accelerometer. It also features a high-performing infra-red camera, allowing it to determine pointing information [Wiili.org, 2009b].

The Wiimote pointing capabilities are described as it “senses light from the console's Sensor Bar, which allows consistent usage regardless of a television's type or size” [Wiire.org]. The Sensor Bar is described to feature “ten infrared LEDs, five at each end of the bar; the LEDs farthest away from the centre are pointed slightly away from the centre, the LEDs closest to the centre are pointed slightly inwards, while the rest are pointed straight forward” [Wiili.org, 2009b].

The precision of the pointing feature is described:

> This composition (of the 10 LEDs) allows an accurate pointing capability to the Wiimote, at up to 5 meters away from the bar. The light emitted from each end of the Sensor Bar is focused onto the image sensor which sees the light as two bright dots separated by a distance "mi" on the image sensor. The second distance "m" between the two clusters of light emitters in the Sensor Bar is a fixed distance. From these two distances m and mi, the Wii CPU calculates the distance between the Wii Remote and the Sensor Bar using triangulation. In addition, rotation of the Wii Remote with respect to the ground can also be calculated from the relative angle of the two dots of light on the image sensor [Wiili.org, 2009a].

The use of an infrared sensor to detect position can cause some detection problems when other infrared sources are around, such as incandescent light bulbs or candles. The solutions for this kind of problem are also discussed:

> This can be easily alleviated by using fluorescent lights around the Wii, which emit little to no infrared light [Wiili.org, 2009a, Wiire.org]. Innovative users have used other sources of IR light as Sensor Bar substitutes such as a pair of flashlights and a pair of candles [Wiire.org]. Such substitutes for the Sensor Bar illustrate the fact that a pair of non-moving lights provides continuous calibration of the direction that the Wii Remote is pointing and its physical location relative to the light sources. There is no way to calibrate the position of the cursor relative to where the user is pointing the controller

without the two stable reference sources of light provided by the Sensor Bar or substitutes.

The high success of the Nintendo Wii video-game is greatly due its gesture/haptic interaction model. Its market penetration proves that people are eager and ready to try new ways to work with machines. While competing machinery would require many hundreds or thousands of Euros, a Wiimote is more accessible, starting at around 45€. This opens a new door of opportunities to exploit and expand the Wiimote's capabilities and allow users to use it in other applications, not only for the gaming.

The device uses the acceleration information to provide a very simple gameplay control on the console. There are titles that use a more complex mapping of gestures into actual commands, such as *Dragon Ball Z*, but the most popular game is the *Wii Sports*, where the user gestures are a direct map of a given sport move. Figure 5 shows examples of that in A, B, and C where users simulate actual boxing movements, tennis swings, and a football head move.

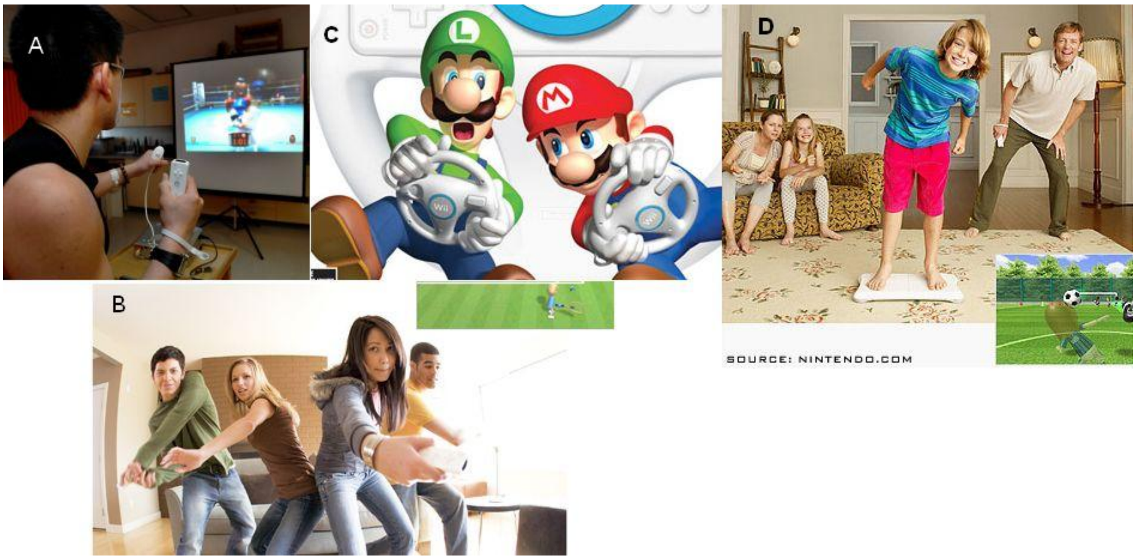

Figure 5 Gesture types

The popularity of *Wii Sports* is an important fact because it clearly demonstrates how users really have no need for (and possibly dislike) more complex gesture interfaces. The example given previously, *Dragon Ball Z*, is one of these examples where gamers are required to "*fake a sorcerer's spell conjuring gimmick*" in order to relay a command to the machine.

6.1. Motion sensing

The 3-axis accelerometers report, in arbitrary units, the acceleration imparted on the controller according to its usage. At rest, acceleration is equivalent to the total force (or weight) of gravity, "g", in the upward direction. In free fall, it should report approximately zero acceleration. Any body in a 3D space has six degrees of freedom, three linear translation directions and three rotation angles, and the Wiimote reports each of these degrees of freedom in arbitrary units, structured in the report communicated via Bluetooth. These values vary in decimals between -1 and +1.

The Wiimote sensing capabilities for the yaw rotation are limited when it's held in the standard position because gravity exerts no force on that axis; therefore the accelerometer reading depends only in the velocity of motion. If a user yaws the Wiimote very slowly, this will not be detected by the accelerometer and the interface could loose reference to the controller's position.

Wiili [2009] discusses options to overcome this problem by using the Wii Sensor Bar as a reference, and using the pointing information from that reference to define the yaw rotation.

6.2. Interaction models/methods/techniques

The Wii game console uses the Wiimote mainly in two ways. At first, when the user is presented with a menu the interaction is controller by using the remote's pointing capabilities (perhaps because when several options are on the screen, there is need for precision and efficiency of control). During gameplay the Wiimote's full power is exploited – here users must combine pointing (usually to provide directionality) with movement cues.

Although these methods are sufficient to provide users with a satisfying range of interaction techniques, it is still possible to develop alternatives that allow the production of new ways to use the Wiimote. There are several examples on the Internet ever since the Wiimote was reverse engineered [Wiili.org, Wiire.org], on propositions of new usages for the Wiimote. The most famous and popular sources of inspiration usually refer to Lee [2008a], one of the first to provide applications and distribute them. His contribution came when proposing to use the Wiimote's camera to track points of interest that have several purposes. On one article Lee explores the possibilities of using the Wiimote to track marked points on foldable displays that have image projected from a standard projector [Lee, 2008a].

Having the Wiimote as a stationed camera requires the interaction to be oriented with infra-red light. The points of interest that must be mapped with the controller must be marked with infra-red light, either by actually emitting the light or being a source of reflection to it. This allows for many techniques that can benefit from tracking, enabling almost any object to become "interactive". Lee has experimented with interactive whiteboards, where the user holds a special pen that emits infra-red light, turning the whiteboard literally into a touch-screen. It has been demonstrated how it's possible to use the Wiimote (for hand-tracking) combined with other equipment to simulate "touchable holography" [Shinoda et al., 2009].

6.3. Limitations

The Wiimote's accelerometer has troubles to properly inform about yaw rotations, the reason for this being that there is practically no gravity force on the axis that is parallel to the ground. In order for the accelerometer to detect changes in the device's orientation in the horizontal plane, the movements need to be sufficiently fast so that they infer some degree of *g* force on the sensor. If a user turns the Wiimote rotating it slowly horizontally, the controller will not be able to report this change because its sensitivity is limited. Yim [2008] had previously reported that inertial tracking is fast and does not suffer LOS problems, but is inaccurate when capturing slow position changes.

One controller can only connect to one host at a time, which means that it cannot be used in a "task switching" manner, e.g. between different machines (multiple computers for instance). It is not a standard to use the Bluetooth radio in promiscuous mode so that multiple hosts could receive reports – even if this was done there would be need for synchronization between the hosts to define which one would interpret a gesture/action and react to it.

The Wiimote's vision is restricted to infra-red light, which can be considered as a limitation. The camera contained in the Wiimote could possibly be used for capture of any sort of image, meaning that it could potentially be a tagging machine or a general purpose camera. If this were the case, a Wiimote's vision could be used for giving applications contextual information about a user's point of interest or his environment in general. Another issue is that there are only four points that can be tracked by the controller at a time. It would be desirable to define how many points the Wiimote should try to track.

6.4. Tests

The suitability of the Wiimote has been proved in many different applications [Bernardes et al., 2009; Bruegge et al., 2007; Lee, 2008a, 2008b, 2008c; Prekopcsák et al., 2008; Shinoda et al.. 2009; Shiratori and Hodgins, 2008; Wiili.org, 2009b]; and here its usage will be evaluated in two situations that aim to investigate the usability of the controller in generic tasks, given that it can be used as a gesture interface controller, and to verify its suitability in day-by-day or intensive use.

If the Wiimote evaluation as a gesture interface controller is realized (independently of the results), we also propose to investigate it in a modality switching manner – changing the interaction between a gesture interface controller into a replacement for mouse and keyboard. With this, we look into evaluating its advantages and disadvantages in comparison to standard keyboard-mouse alternatives.

6.4.1. Wiimote for pointing and text input

The Wiimote will be used in tandem with an ordinary computer, as a mouse. The users will be given tasks that resemble activities performed daily, such as Internet browsing and writing e-mails. The Wiimote can be configured as a mouse in Windows or Linux settings, but this test will focus on Windows environments. The keyboard functionality for text input can be provided by any virtual keyboard technology, which has been extensively used and investigated for text entry in many different scenarios where keyboards are unfitted (tablet PCs, mobile phones).

The aim of this test is to evaluate how users perform with the Wiimote, not to compare the efficiency with normal interaction methods such as keyboard-mouse combinations. This test will also evaluate how the Wiimote can integrate with other solutions for HCI alternatives.

In practical terms, a managing program must be created to monitor when text input is needed, offering users with a virtual keyboard's interface.

At the end of the experience, users will answer the following questions:

- how comfortable do you feel to use the Wiimote as a mouse;
- how efficient would you evaluate the control of Windows with Wiimote;
- how tiresome is it to use the Wiimote as a mouse;
- how responsive was this interaction;
- how productive do you think this experience was to perform your tasks;
- rate the overall experience.

The users would receive explanations of what is efficiency and productivity in this interview. Before the experience starts, some users would also receive brief introductions of what is intention of the study. These users would also be told that the research doesn't expect them to perform fast or quickly and that the interaction is expected (in advance) to be restricted – some things like mouse-keyboard shortcut combinations will just not be possible. The answers to these questions are ranked in a scale from 0-5.

Apart from the discrete questions, the users would also be asked for open feedback:

- If you have a Wiimote at home, would you like to set it up with your computer?
- Do you believe that the Wiimote can be used (at least in some situations) to replace keyboard and mouse?

6.4.2. Technology applicability and interface feasibility

As previously mentioned, the intention of this thesis is not to innovate in the recognition of gestures – the Wiimote will be used for basic gesture control without gesture interpretation. In the field of HCI most research focuses on semantically basic gestures [Gunes et al., 2004]. Even this task has presented itself as a challenge for most of the reviewed literature. More interestingly, in order to really contribute for an intelligent computer-human interface, it is necessary to use and understand emblematic or even more complex gestures [Gunes et al., 2004]. This would be a topic for another more extensive research, though.

To test the Wiimote against basic gestures, an application has been developed. It uses the Wiimote as a camera, tracking two of the users' fingers. They are used to "grab", rotate, and move items in the screen. This application will aim to test a new interaction technique where users wave their hands in the air to control objects in the computer screen.

This application is based on a proposition by Lee [2007]. It uses the same previously mentioned library to retrieve the controller's information. The users are given a task to review a series of photos that may need to be rotated; when finding a picture that is in the wrong orientation, users issue a command to rotate them in the correct direction (clock-wise or counter-clock-wise) by swirling the index finger and the thumb. It is possible to advance or retrocede back to next or previous pictures by moving the hand quickly to left or right. A possible scenario for this use would be a photography kiosk.

The setup is such where the user is located right in front of the screen, at about three meters from it. Under the screen the Wiimote is placed, directed upwards towards the area where the user will gesticulate. Above the screen there is an infra-red light emitting array of LEDs, connected to a power source – it's always on and this is the source of IR-light reflection for the Wiimote "vision". The light coming from the array will reflect on the users' fingers, bouncing back in the direction of the Wiimote. In order to improve resolution and precision, users should wear a reflective material on the tips of the fingers, which will help the Wiimote differentiate them from the rest of the hand (which also reflects the IR-light).

The implementation of this program is done in C# Language, using Microsoft Visual Studio 2008 Express edition. It also uses some of Microsoft DirectX technologies to read and write from the HID devices.

After using this program for at least 10 minutes, every user will answer a survey. This interview will investigate if this technique is suitable for the task at hand; if it is usable; and if the detection and tracking is sufficiently efficient. The interview will contain the following questions:

- How well the computer understood your commands;
- How efficient was this technique;
- How productive is it to perform this task with this technique;
- How tiresome;
- How intuitive/natural is this technique.

Then in the descriptive questions, users would be required to answer:

- Would you like to see this technique implemented in other appliances?

6.4.3. Development case

Since the Wiimote can be treated as a HID device, most of today's operating systems can support communication with it – the already available applications (for Windows, Linux, and Mac) are numerous. The design of applications can be envisioned generically in two layers: the bottom layer would implement the wrapping of communication, exposing the Wiimote's properties and raw data; the higher layer containing the application business, the interaction management per se.

Several efforts have been made to provide sources of a base framework for the Wiimote connectivity and interfacing; for this project Brian Peek's solution was used – it is a .Net library developed in C# that supports notifications via events. The WiimoteLib has a few limitations, however: one considerably serious issue is that is related with the way it blocks and waits for responses from the Wiimote in a synchronous way. This can cause problems when the Wiimote events are directly handled (bound) to the user-interface layer of the application, because the subsequent calls caused by user-interaction could prevent the library from receiving reports from the Wiimote. This library must be redesigned so that it operates in multi-threaded mode, making asynchronous calls to deliver report data to its clients.

Developing an application in C#, based on similar libraries is considerably fast. The creation of a library itself for communicating with the Wiimote is not a complex task, given that the reverse-engineering of the device has already been completed. A developer with reasonable level of understanding of .Net can ramp-up a program within a month. The application subject of this study was developed in Microsoft Visual Studio 2008 Express Edition, using C# and the Windows Presentation Foundation class-libraries. The approximate number of hours used for development and investigation is 640 hours.

This approach is based on the suggestions of Lee [2007, 2008a]. It requires a setup that is, to say the least, unorthodox: the Wiimote, contrary to normal usage, is placed away from the user. The controller must be placed in a way that it can "watch" the user as he/she uses the hands posed in front of the torso or head. The idea is that the Wiimote must be able to see the person as he/she stands in front of the screen.

Another piece of hardware must be acquired in order to complete the setup: an infra-red LED-array (see Figure 6), which will produce a strong source of IR light to be reflected from the users' fingertips back into the Wiimote camera.

Figure 6 Infra-red LED-array

The system should be used in such way that the light emitted from the IR-array reflects on the users fingers back to the Wiimote field of view, as demonstrated on the following image. The IR-array is now active and coloured in red, on top of a TV.

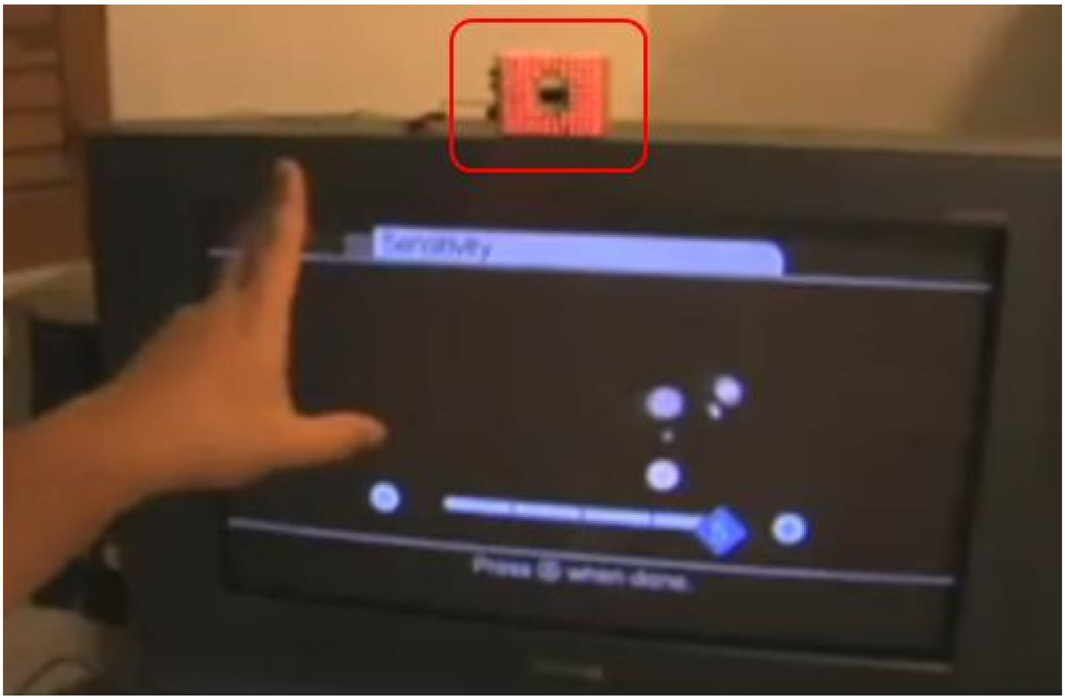

Figure 7 Activated IR-array, system function

In Figure 7, the IR-array is used in combination with the Nintendo Wii's sensitivity calibration program, displayed on the TV screen. This is perhaps the most important component of the whole system, since the interface depends on the reflection of the IR-light on the users' fingertips. Due to the essential condition of this item, it must be readily available for purchase and/or assembly to the great majority of the users.

An important fact to note from Figure 7 is the noise visible in the TV, result of the reflection of infra-red from the user’s hand. This noise can ultimately turn the system infeasible, it is necessary to eliminate or neutralize it.

As suggested by Lee [2008a], one could assemble the IR-array by using parts found on most electronics stores. This, however, is not a suitable solution for ordinary users – it is necessary to have commercial solutions available for purchase. Fortunately there are several manufacturers of devices used in surveillance systems, which resemble IR-arrays needed by this interface.

For the development of the suggested application it was clear the necessity to reproduce conditions of real users. In order to do so, a commercial IR-array has been purchased to conduct a series of tests and develop an initial prototype application to try out the viability of the system.

During the first checks, it was noted that the Wiimote could produce responses to the interaction. These initial tests succeeded to report data from the controller’s sensors and the system even managed to communicate with device extensions (i.e., Nunchuck). The necessary communication was established and the data was successfully being extracted from the device and transmitted to the application.

The performance of the system in regard to the usage of infra-red light for positioning points in the screen was seriously poor. Firstly, using no additional hardware simply doesn’t work because the system is easily confused on which fingers to track (the system can also mistakenly capture reflections on the user’s hand – knuckles and palm). So, in order to properly select a point on which to focus the tracking, it would be necessary to highlight it, or mark it. A solution was proposed by Lee [2008a]: reflective tape (catadioptric reflectors). This solution does not work with most commercially available IR-sources. Lee [2008a] built an IR-LED array using a design where the outer LEDs point outwards; while the inner LEDs point inwards. This makes it easier for the light to be reflected from different angles. In commercial IR-sources the light is more focalized.

Retrieving the infra-red light data proved to be inadequate: the raw data would, most of the time, report flicker of locations (coordinates that were not adjacent). Clearly there was a problem in the resolution – the Wiimote needs a finely adjusted IR-array that can produce a considerably strong source of IR-light, strong enough to reflect on a user’s fingertips which could be located at a far distance.

The “flickering” mentioned above was, in fact, an alternation of the fingertip’s coordinates (as if the user would rapidly move his finger to another place). The cause for this problem is the IR-light being reflected by multiple points on the hand. Even with an alternative IR-source it has been proven difficult to resolve this issue.

Different approaches were examined, from changing the materials that were worn on the fingertips, to adjustments on the IR-sources, but there was little improvement and the flickering persisted in all cases. It's been also experimented placing the Wiimote on different distances, adjusting the delta between Wiimote-IR-source-fingertips; this however didn't help – the interface would either completely lose perception or keep presenting oscillations on the fingertips' coordinates.

Instead of trying to fix the problem at the source, a proposition has also been examined on the application-level by implementing software normalization for the oscillations. The issues concerning this solution are discussed next.

The proposition of software normalization would be reasonable only if the interface was to be used with applications that complied with the concept of "10-foot design". The reason for this is that normalizing the oscillations would cause each fingertip to be "seen" as an area that is bigger than it is in reality – the result is that users could be misled by bad precision on pin-pointing their gestures. The image below shows this normalization.

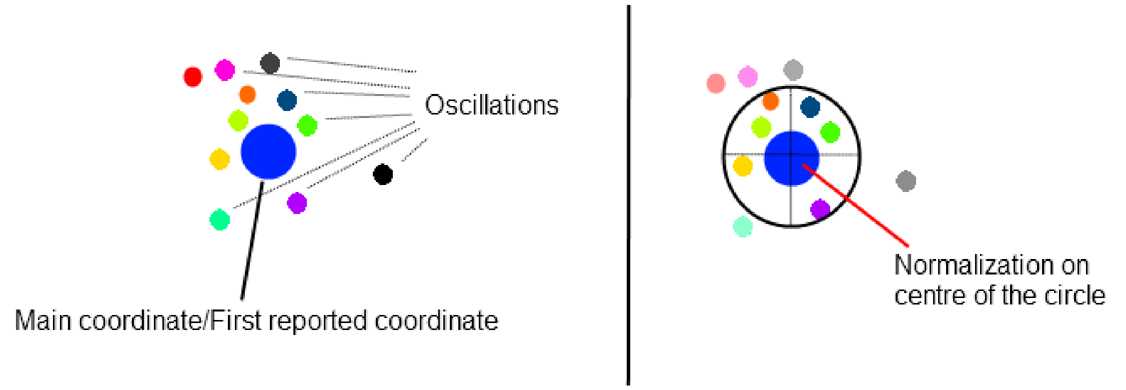

Figure 8 Normalization

The normalization depicted in Figure 8 demonstrates how some oscillation of coordinates would be filtered out of the system and how the mean would be calculated taking in account only the valid coordinates. This represents an error tolerance of 10 – 40 pixels, when the actual size for the user finger would be equivalent to 10 pixels. In practice this means that the applications using this interface need to make sure that no more than one user-interface control is positioned in an area of 40 pixels.

This is a problem in every setup that is not a 10-foot or where the interface uses many UI controls such as a generic desktop setup. Because ordinary computers use small elements (every interface nowadays relies at least on menu-systems with 16x16-pixel buttons) the solution presented here will not make the whole system suitable for the largest range of users.

Due to these facts the implementation of any similar interface, that places the Wiimote away from the user and attempts to read infra-red light reflections, cannot be guaranteed to work well. Firstly, it is necessary to find a manufacturer of IR-light sources which focuses specifically on the requirements of the Wiimote and on this proposed approach to gesture interfaces. Perhaps then, with a standard and guaranteed

IR-light array the feasibility of the whole system can be ensured. Secondly, the challenging task of achieving a satisfying reflection from the users' fingertips has to be accomplished; otherwise it will be hard to make good use of any IR-light source.

It is clear that these necessities eventually result in the situation where a gesture interface solution which was supposed to be generic and simple, becomes considerably restrictive – because it requires additional equipment (hardware) which is not simple to assemble nor easy to find in ordinary stores.

This indicates that other technologies might be more suitable to the development of similar interfaces. One possible solution would be the Xbox Natal project, which is still to be launched – it has been reported to perform at an excellent level when recognizing and tracking the whole body of the user.

The results of the present work are discussed next.

7. Evaluation and Discussion

There are several applications for the Wiimote in desktop computer setups. Software is already available permitting technology enthusiasts to apply the use of this device on their daily lives. The application to HCI purposes has also been tested, providing important information on the performance and feasibility of its use.

This work started with a very strong link to the general population, computer users from all over the world. The aim was to find alternatives for providing this public with the opportunity to experience interfacing the computer through gestures.

Some goals of this research have been met: after a careful review of previous contributions to the field of HCI, thorough analysis of technological aspects and issues presented by other researchers, it was possible to identify certain qualities that make a technology adequate for the goals of the present thesis. The definition of the *ideal technology* is a base framework upon which the next step of this research developed: the Wiimote was identified as the best potential candidate to assist in creating gesture interfaces that would be accessible to the general public.

The literature review also provided valuable material for the development of the next part of this work. Other researchers have already examined some applications and potentials of the Wiimote, most of the content are positive to the use of this device. Inspired by a research proposition for future work [Lee, 2008a], this thesis proposed a design that could combine multiple hardware parts in order to compose and complex setup that would allow for hands-free gesture interaction. During the analysis of the Wiimote technology and development of the proposed design several shortcomings were confirmed, matching observations made by other researchers.

The Wiimote technology has been praised for allowing for many different kinds of interactions. The standard configuration (Wiimote in hand, interface in front of the user) allows remote-control of the interface, similar to a wireless mouse. It also permits multi-user interaction (as long as the computer software counterpart supports it as well). It can allow for shared display interactions; and can even be used to track virtually any generic infra-red object (an object which emits or reflects infra-red light), serving as tag scanner or locator.

The proposition presented here is a shift from the original approach applied by the Wiimote – the controller combined with an IR-light source are placed away from the user, (in the same plane) “watching” the user’s fingertips.

Although a prototype has been developed and demonstrated [Lee, 2008a], this served only as a technical proof of concept. The demonstration was developed by using a custom-made IR-light array, and while similar to the commercial models, it is not standardized.

The design presented by that prototype has originated the setup, concept and design presented here. The initial idea to invert the Wiimote usage and place to controller away from the user seemed to have great potential since any equivalent alternative would have elevated costs, while the Wiimote solution is cheap (even with the extra electronics required for a complete environment).

It has been found that this approach used to a gesture interface with the Wiimote does not always work because of variations caused by the additional components of the system – the combination of non-standardized reflective material and non-standardized IR-light sources make up for a very fragile solution.

Apart from the problems related to the design and style of the interface, several other criticisms can be made to the approach proposed for this gesture interface. The first is related to the IR-light source, because such setup where the user is required to remain in a determinate area (field of reach of the IR-light) brings to the interface the same handicaps as the ones present in Computer-Vision based interfaces. In this sense it would not benefit from the advantages of movement and freedom of positioning, inherent to generic Wiimote usage. This could only be achieved if the users would wear active infra-red elements (i.e., IR-LEDs).

In addition, one can combine the previous issue with the Wiimote's camera field of view restrictions, which is also influenced by the camera's sensitivity. The device simply cannot "see" more than 45 degrees of field, and even if the user is positioned directly in front of the controller the sensitivity of the camera imposes restrictions as to how far the user can be in order for the IR-light to be reflected with enough strength to travel back to the Wiimote.

These two characteristics together make the whole solution quite complex to be understood by ordinary users, because they must have the comprehension of how one part relates to the other (while not being able to see the infra-red light). Although it is usually enough to place one item beside the other (having them both point towards the user), users will most likely have trouble configuring their environment at the best possible efficiency so that the infra-red light reflection is well captured by the controller.

Since humans cannot see infra-red light, there is potential trouble on how to configure the angle at which each part is posed – it is even hard to tell if the system if on or not.

Another problem to the design is that the reflection of the IR-light on the users' fingertips can vary greatly. This could be arranged by requiring the users to restrict their location within the field of view of the Wiimote; but there would still be problems when used by children, whose surface area on the fingertips is significantly smaller than an adult.

The solutions to the restrictions of positioning and signal strength are also not optimal – requiring users to wear gloves or any sort of special reflective material on the fingertips reduces the general usability of the whole system. The ideal situation would such where the users don't have to prepare themselves to use the interface.

This does not mean however that the Wiimote is not the best option to the first goal of this thesis: to find the best possible candidate which would permit the development of gesture interfaces to the general public. As it has been demonstrated in the previous chapters, this device is still one of the most remarkable options because of the standard mechanisms of communication to the computer, it's embedded hardware and pre-processed information, and also because of the implementation of a standard HID device profile. As the characteristics and properties which make this device extraordinary have already been discussed in previous chapters, they will not be reviewed now.

Other researchers have succeeded in developing solutions using the Wiimote, which proves that it is fit for diverse applications and domains. Bernardes et al. [2009] succeeded in integrating the Wiimote with an existing gesture engine. The first and most important issue in their work was the controller's refresh rate (which proved to perform at 100 MHz), making it possible to use the device without further problems.

Their group developed a research based on observation of the controller's behaviour when used; they analyzed the reports provided at each movement and afterwards configured pieces of software to interpret the data, and used those devices the game engine *enJine*. The goal of these tests (or the result of the research) was to find how much implementation effort would be required in order to integrate the Wiimote, and the positive result indicated that a small amount of work is required.

Bruegge et al. [2007] have also succeeded in using the Wiimote as a virtual baton, having reported no considerable problems. It is interesting to note that in their work they used the Wiimote and another similar device which also provided acceleration data, and both performed similarly. The extensibility and scalability of the Wiimote technology is proved by Lee et al. [2007] when demonstrating a hybrid system for infrared and visible light projection for location tracking.

The type of custom use-cases (or interface design) where the device is used in a completely new approach, such as the one investigated in the present work, can indeed work. Although the one presented here has failed due to the goals set for this thesis and the complex systematic, others have succeeded in creating rather revolutionary solutions.

Work by Lee et al. [2007] is one example where the use of the Wiimote is similar to the proposition made here; with the difference that the points of interest in the tracking operation are active (they emit light instead of reflecting it). This fundamental difference may be the one single factor of success; but it would require users of the system proposed here to wear hardware, which has been discussed to seriously reduce usability.

The use of the Wiimote and this approach is also applicable to other concepts too, such as the one presented be Lee et al. [2008b] where a system of foldable displays is presented. Using the same technique of tracking active infrared points of interest, the team manages to recognize projection surfaces that can change shape, like a foldable fan, an umbrella, etc. This is a proof that the same method of interaction with the Wiimote can be used in different designs, given that all the variables are under control and deviations or variations are tolerable.

These two are examples of unorthodox usages of the Wiimote, but not only eccentric designs can be produced with it. While it has been endorsed here that gesture interfaces should be simple, easy to learn, and efficient; Schlömer et al. [2008] have accomplished just that. They used the controller to develop generic gesture recognition method based on Hidden Markov Models. They proof that it is also possible to use the controller with simple gestures such as ones that resemble a square, a circle, or a tennis swing. It has been discussed that these gestures do not instinctively recall any particular command to most users; but nevertheless this work shows that the recognition of gestures can be abstracted from the device used, meaning that the Wiimote could possibly be used with other methods as well.

The revolutionary kind of design proposed here, and those idealized by Schlömer et al. [2008], Shiratori et al. [2008], Shreedharan et al. [2007], and Wong et al. [2008] have at least one notable serious drawback. As reported by Shreedharan et al. [2007], these adaptations of interface designs result in users being unfamiliar, often devising ergonomically inappropriate gestures for the interaction. The problem with mapping gestures into computer commands is that users may use similar gestures for very different ways in different contexts.

Not only the issues related with interface design and application concepts are limiting factors in this field. As we rely on stratum that is out of our scope of concern, we abstract complex issues out of the system, assuming that a determinate component or infra-structure is a given. That is the case when dealing with Wiimote-based interfaces – researchers trust that the operating system and device manufacturer will provide necessary resources for the system to work.

Even in recent literature such as Bernardes et al. [2009], the dependency on a good Bluteooth-radio library and underlying system infrastructure can be a problem. Although in most Windows setups this may not be a common problem, there is still not a standard way to solve the issue – the Bluetooth device driver providers are still the main responsible party for them working well.

The particular issue of tracking users' fingertips was the central problem in this work, but Lee et al. [2007] have achieved a rather flexible and robust system that can track points of interest on a surface with considerable accuracy. Even while the user manipulates the surface changing its tilt and inclination, the tracking presents remarkable performance. In order to realize such similar system, users of the present work would have to be required to wear active IR-light diodes on their hands.

In the present work it has been found that the envisioned design cannot be realized unless significant modifications are made. It is estimated that this design cannot be developed with any other currently available device either. There exists, in the field of Computer Vision, examples of gesture interfaces which are based on normal camera and picture analysis, but the shortcomings and problems (which have also been presented in previous chapters) of these alternatives make them no better than the Wiimote-based solutions.

Not only the hardware and software must be integrated with success when creating a gesture interface – the technical feasibility might not be the final limiting factor after all. Shiratori [2008] dully refers to this fact when designing and testing an interface: while focusing on using the commercially available Wiimote they leveraged the potential of the interface because in doing so, they removed possible obstacles in the creation of a complete product chain. Research rarely possesses the power to implement the whole chain of product development, marketing, production, and sales – a cycle that is almost basic and fundamental in order to provide for consumers.

8. Summary

There is a plethora of devices on the market, or coming in the next months, which enable users to interact with machines via gestures. They range from powerful game consoles with many specialized processors and large amounts of memory, to smaller and more restricted telephones. Almost each example implements their own interface model, having little to share with each other. This fact causes an increase in the cognitive load to users who must learn how each one works in their particular way.

These machines enable gesture interfaces by using several types of techniques and technologies – some use an infra-red light spot and track movement, some project structured-light on their subject of interest. While this proves that there are many ways to achieve the same (or similar) result, the differences between the available interfaces make it possible to devise requirements for a common level of quality to gesture interfaces.

This thesis started with an important aim, to find ways in which new interaction techniques could be made available to the widest population; stretching the reach of the HCI field to a massive public. After reviewing previous research and considering challenges and solutions reported by other researchers, it was concluded that it was necessary to analyze needed characteristics and properties that a technology would need in order to provide for this purpose. As a subsequent step to this analysis, one device in particular was identified, which can best accomplish the desired role.

The Wiimote is a powerful device with great potentials to assist in the delivery of solutions with new interaction techniques. It allows the development of revolutionary applications in terms of HCI, because it permits freedom of movement while maintaining precision and robustness. It is undeniable how well received and accepted it is by its users, and how natural they feel using it; it appears that all the sensation in the shops is actually explained.

The design and implementation of the original interface are satisfying and pleasant to use, but there are still cases where it is felt that the machine does not properly "understand" what users mean – the Wiimote is not perfect. As it has been exposed previously, the Wiimote has flaws and limitations that make it a device suited only to a certain number of applications. It cannot, for example, be used in tasks where it is necessary to perform extremely precise operations: the Wiimote is a tool for gestures and pointing, not for delicate interactions.

Although this device is limited, the combination of all the met requirements makes it an unchallenged candidate. In order to verify the theory, it was necessary to try to apply this technology to a generic gesture-based interface. This could indicate the suitability of the controller for more diverse applications as well.

Even if this device is limited to a determinate amount of usage cases, it is still to this date the one single piece of technology that offers most possibilities for the least price. If the objective is to develop gestural interfaces or applications that target the largest possible audience, the Wiimote is the platform for the solution. The Wiimote's scalable engineering allows for its subsequent updates, like the pseudo-guitar extra, the balance-board, and the "extra mobility pack" which gives even more precision to the controller. This means that we are still to see all of this device's potentials; and is a clear indicator that more applications may become possible.

As a continuation to future work proposed by other researchers, the design introduced by the present work was a non-standard utilization of the device. What made the design unique is the fact that it actually permits users to wear no equipment on themselves, and allows a considerable level of flexibility, mobility, and freedom of movement. The nature of the technology also results in the benefit of it being practically impervious to external interferences such as changes in light conditions, or other environmental interferences.

The selected approach has revealed several problems during the implementation phase of a test application. The solution is not viable to be used by most users because setting up the system can easily become a major problem – there are complicated parts to manage. Combining an infrared-light array with reflective tape turned out to be cumbersome and not enough user-friendly. The reflection was not always precise and constant, which severely compromised the efficiency and robustness of the whole experience.

Although innovative and to some extent a revolutionizing interface model, it proved to be inadequate for the Wiimote. The dependencies to external functional objects such as the IR-light source and reflective qualities of the users' fingertips became difficult issues to coordinate and manage.

The functional role of the IR-light source proved unstable because of unpredictable conditions for the reflections from the users' fingertips. As a counterpart to the IR-light source, the users' fingertips revealed a problem with the insufficient intensity of the IR-light, to which a solution was proposed – increment the system with the use of materials with better reflective properties than the human skin and tissue. This configuration, however, failed to provide for the needed performance, as the perception achieved with the Wiimote would oscillate the position of the users' fingers in an unacceptable scale.

One particular problem faced during tests is that the standard Bluetooth stack bundled by Microsoft in Windows systems does not provide mechanisms to initiate a connection with the Wiimote. Users are able to search for and pair their computers with the remote, but after the first configuration, it is impossible to open the connection with the controller. If there are no other tools that can facilitate on this task, it will be necessary to remove the Wiimote from the paired devices list, and add it again. Opening this connection depends entirely on an application that must be created by the interface creators. The reason why this is required is that the libraries utilized do not contain any Bluetooth-related algorithms; they simply try to open a channel to HID input device, unaware of the underlying physical link to the controller. If the Wiimote is not connected, there is no such HID device on the system. In other words, the radio connectivity is transparent to the library software.

The complexity introduced by the IR-light source and the reflection on users' fingertips finally rendered the system infeasible. The design proposed by the present work could not be realised, nevertheless the Wiimote is still the best hardware candidate to assist the field in disseminating gesture interfaces to the general public. It is the cheapest device available in the most countries, even when compared to upcoming products such as the new Sony Playstation Move controller. It contains state-of-the-art technology using components that are relatively simple and have been available for years. In addition, it operates (even if unofficially) on open standards that are supported by most computers of today.

This thesis features several conclusions about the issues concerning the study, most importantly related to the identification of a powerful technology, its limitations, and a generic use of it. It has been found that even with such excellent device, it is extremely important to design an interface that is compatible with the device's native capabilities. Trying to extend and expand the device's power and function is likely to fail on several occasions especially when the expansions and extensions do not comply with the initial design of the device itself.

The results verified with the developed application show that the Wiimote is not a device that is compatible with the interface model proposed here. There is a myriad of interaction styles on which to apply the Wiimote technology, but perhaps the one devised here requires a different kind of technology to be realized. One option for future work would be the attempt to realize this with use of the Xbox Natal project, which might contain enough embedded processing resources to abstract the tracking complexity.

References

[ACM] SIGCHI Curriculum for Human Computer Interaction - http://sigchi.org/cdg/cdg2.html#2_1

[Baudel and Beaudouin-Lafon, 1993] Thomas Baudel and Michel Beaudouin-Lafon, Charade: remote control of objects using free-hand gestures. *Communication of the ACM*, **36**(1993), 29-35.

[Bernardes et al., 2009] João Bernardes, Ricardo Nakamura, Daniel Calife, Daniel Tokunaga, and Romero Tori, Integrating the Wii Controller with enJine: 3D interfaces extending the frontiers of a didactic engine. *ACM Computers in Entertainment*, **7**(1) (2009), 12:1-12:19.

[Bruegge et al., 2007] Bernd Bruegge, Christoph Teschner, Peter Lachenmaier, Eva Fenzi, Dominik Schmidt, and Simon Bierbaum, Pinocchio: conducting a virtual symphony orchestra. *ACE'07*, **6**, 294-295.

[Cabral et al., 1995] Marcio C. Cabral, Carlos H. Morimoto, Marcelo K. Zuffo, On the usability of gesture interfaces in virtual reality environments. In: *ACM CLIHC'05*, 10-108.

[Grossman et al., 2004] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan, Multi-Finger Gestural Interaction with 3D Volumetric Displays, In: *ACM UIST 2004*.

[Gunes et al., 2004] Hatice Gunes, Massimo Piccardi, Tony Jan, Face and body gesture recognition for a vision-based multimodal analyzer. In: *Pan-Sydney Area Workshop on Visual Information Processing (VIP2003)*, 19-28.

[Hemmert, 2009] Fabian Hemmert, How Can We Make Digital Content Graspable? At TEDxBerlin on November 30th, 2009.

[Kallio et al., 2006] Sanna Kallio, Juha Kela, Jani Mäntyjärvi, Johan Plomp, Visualization of hand gestures for pervasive computing environments. In: *AVI'06*, 480-483.

[Kang and Ikeuchi, 1994] Sing Bing Kang, Katsushi Ikeuchi, Determination of Motion breakpoints in a task sequence from Human Hand Motion. Carnegie Mellon University, Robotics Institute, Report 1050-4729/94, 1994.

[Kang and Ikeuchi, 1994] Sing Bing Kang, Katsushi Ikeuchi, Toward automatic robot instruction from perception - temporal segmenattion of tasks from human hand motion. *IEEE Transactions on robotics and automation, **11** (5)*, 1995, 670-681.

[Kela et al., 2005] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Guiseppe Savino, Luca Jozzo, Sergio Di Marca, Accelerometer-based gesture control for a design environment. *Pers Ubiquit Comput*, **10**(2005), 285-299.

[Latoschik, 2001] Marc Erich Latoschik, A gesture processing framework for multimodal interaction in virtual reality. In: *Afrigraph 2001*, **01**, 95-100.

[Lee et al., 1998] ChanSu Lee, SangWon Ghyme, ChanJong Park, and KwangYun Wohn, The control of avatar motion using hand gesture. In: *VRST'98*, **0011**, 59-66.

[Lee et al., 2007] Johnny Chung Lee, Scott Hudson, and Paul Dietz, Hybrid infrared and visible light projection for location tracking. Unpublished manuscript, March 2007, available as www.johnnylee.net/academic/hybrid.pdf

[Lee, 2007] Tracking fingers with the Wiimote - http://www.youtube.com/watch?v=0awjPUkBXOU

[Lee, 2008] Johnny Chung Lee, interview for TED on February 2008 - Johnny Lee demos Wii Remote hacks.

[Lee et al., 2008] Johnny Chung Lee, Scott E. Hudson, and Edward Tse, Foldable Interactive Displays. In: *UIST'08*, **10**, 287-290.

[Lee et al., 2008] Hyun-Jean Lee, Hyungsin Kim, Gaurav Gupta, and Ali Mazalek, WiiArts: creating collaborative art experience with WiiRemote interaction. In: *TEI'08*, **02**, 33-36.

[Lee, 2009] Michael Lee, Sony gets dusted. In: *The Red Herring*, 1080076X, p2-2.

[Lenman et al., 2002] Sören Lenman, Lars Bretzner, Björn Thuresson, Using marking menus to develop command sets for computer vision based hand gesture interfaces. In: *NordCHI*, **10/02**, 239-242.

[MacDonald, 2009] Nancy MacDonald, The Season's must-have can't-have. In: *MacLeans.ca*, **121**(51/52), 84.

[Mäkinen, 2007] Erno Mäkinen, Face Analysis Techniques for Human-Computer Interaction. Ph. D. Thesis University of Tampere, Dept. of Computer Science.

[McNeill, 2009a] David McNeill, Gesture, gaze and ground – www.sprinteronline.com/lncs.

[McNeil, 2009b] David McNeil, Gesture. In: *The Cambridge Encyclopedia of Language Sciences* – http://www.cels.uconn.edu.

[Microsoft, 2010] Microsoft Natal Project Home-page. In: http://www.xbox.com/en-us/live/projectnatal.

[Morency and Darrel, 2006] Louis-Philippe Morency and Trevor Darrel, Head gesture recognition in intelligent interfaces: the role of context in improving recognition. In: *IUI'06*, **0001**, 32-38.

[Morris et al., 2006] Meredith Morris, Anqi Huang, Andreas Paepcke, and Terry Winograd, Cooperative gestures: multi-user gestural interactions for co-located groupware, In: *ACM CHI 2006*.

[Naik et al., 2006] Ganesh R. Naik, Dinesh Kant Kumar, Vijay Pal Singh, and Marimutu Palaniswami, Hand gestures for HCI using ICA of EMG. In: *Conferences in Research and Practice in Information Technology (CRPIT)*, **56**, 67-72.

[Nickel and Stiefelhagen, 2003] Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In: *ICMI'03*, **03**(0011), 140-146.

[Nielsen, 2010] Jakob Nielsen, *Tenc usability heyristics*. In: http://www.useit.com/papers/heuristic/heuristic_list.html.

[Nielsen et al., 2003] Michael Nielsen, Moritz Störring, Thomas B. Moeslund, and Erik Granum, A procedure for developing intuitive and ergonomic gesture interfaces for man-machine interaction. Aalborg University, Laboratory of Computer Vision and Media Technology, Report CVMT 03-01, 2003.

[Prekopcsák et al., 2008] Zoltán Prekopcsák; Péter Halácsy, Csaba Gáspár-Papanek, Design and development of an everyday hand gesture interface. In: *MobileCHI 2008*, 479-480.

[Schlömer et al., 2008] Thomas Schlömer, Benjamin Poppinga, Niels Henze, Susanne Boll, Gesture recognition with a wii controller. In: *TEI 2008*, **2**, 11-14.

[Shinoda et al., 2009] Takayuki Hoshi, Masafumi Takahashi, Kei Nakatsuma, Hiroyuki Shinoda, Touchable holography. In: SIGGRAPH09

[Shiratori and Hodgins, 2008] Takaaki Shiratori and Jessica K. Hodgins, Accelerometer-based user interfaces for the control of physically simulated character. *ACM Transactions on Graphics*, **27**(5), 123:1-123:9.

[Sreedharan et al., 2007] Sreeram Sreedharan, Edmund S. Zurita, and Beryl Plimmer, 3D input for 3D worlds. In: *OzCHI 2007 Proceedings*, 227-230.

[Strachan et al., 2004] Steven Stratchan, Roderick Murray-Smith, Ian Oakley, and Jussi Ängeslevä, Dynamic Primitives for Gestural Interaction. University of Glasgow, Dept. of Computer Science. In: *TEI 2008*, **02**, 11-14.

[Wang et al., 2007] Xiying Wang, Xiwen Zhang, and Gouzhong Dai, Tracking of deformable human hand in real time as continuous input for gesture-based interaction. In: *ACM IUI'07*, **0001**, 235-242.

[Wexelblat, 1995] Alan Wexelblat, An approach to natural gesture in virtual environments. In: *ACM Transactions on Computer-Human Interaction,* ***2****(3), (1995) 179-200.*

[Wickey and Alem, 2007] Aiden Wickey and Leila Alem, Analysis of hand hestures in remote collaboration: some design recommendations. In: *OzChi, 28-30.*

[Wilson and Oliver, 2003] Andrew Wilson and Nuria Oliver, GWindows: Robust stereo vision for gesture-based control of windows. In: ACM ICMI'03, 211-217.

Wikipedia – Human-Computer Interaction – http://en.wikipedia.com/Wiki/HCI.

Wiili.com – Wii technology community contributed online resource pages.

Wikipedia – The Wii Remote http://en.wikipedia.org/wiki/Wii_Remote

Wiili.org – how to connect Wiimote with Linux http://www.wiili.com/index.php/Wiimote_linux_tutorial

Wiire.org - Wiimote reverse engineered. http://www.wiire.org 2009

[Wong et al., 2008] Elaine L. Wong, Wilson Y. F. Yuen, and Clifford S. T. Choy, Designing wii controller – a powerful musical instrument in an interactive music performance system. In: *MoMM2008*, **11**, 82-87.

[Yim et al., 2008] Jeffrey Yim, Eric Qiu, and T.C. Nicholas Graham, Experience in the design and development of a game based on head-tracking input. In: *Future play 2008*, 236-239.