Advanced Computer Vision-Based Human Computer Interaction for Entertainment and Software Development

RAUL HERRERA ACUÑA

Submitted in partial fulfilment of the requirements of

Kingston University for the degree of

Doctor of Philosophy

Digital Imaging Research Centre

Kingston University

JULY 2014

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

3DUI	3D User Interface.
AR	Augmented Reality.
CRT	Cathode Ray Tube.
DEViL3D	Development Environment for Visual 3D Languages.
FTIR	Frustrated Total Internal Reflection.
FTV	Free view-point TV.
GIS	Geographic Information Systems.
GPS	Global Positioning System.
GPU	Graphical Processing Unit.
HCI	Human Computer Interaction.
HCII	Human-Computer Intelligent Interaction.
HMD	Head-Mounted Display.
IDEs	Integrated Development Environments.
IMDB	Internet Movie Database.
LEDS	Light-Emitter Diodes.
MERL	Mitsubishi Electrical Research Laboratory.
MRI	Medical Resonance Image.

OLAP	On-Line Analytical Processing.
PCs	Personal Computers.
PESs	Pose Estimation Systems.
PyMT	Python Multi-Touch.
RbD	Robot Programming by Demonstration.
RGB	Red, Green and Blue.
SAM	Solid Agents in Motion.
SDG	Single Display Groupware.
SDK	Software Development Kit
SQL	Structured Query Language.
TUIO	Touch User Input/Output protocol.
TUIs	Tangible User Interfaces.
UML	Unified Modelling Language.
VPL	Visual Programming Languages.
WIMP	Windows, Icon, Menu, Pointing, device.
XCDL	XML-Oriented Composition Definition Language.

Symbols

ac_k	Unbiased acceleration.
ac_k^{in}	Acquired data from the accelerometer in each axis.
b_k	Estimated bias.
q	Orientation quaternion provided by the sensor fusion data.

ac	Readings coming from the accelerometer.
gc	3D vector that represents the dynamic acceleration.
v	Velocity.
a	Acceleration.
s	Position.
X	X axis.
Y	Y axis.
Z	Z axis.
T_{cm}	Transformation matrix.
p	Point on 3D virtual object.
ω	Rotation angle.

Abstract

Human-Computer Interaction (HCI) is a multidisciplinary research area that seeks to improve the methods to interact with computers, improving the technologies in hardware and software to create interfaces capable to resemble the real world interaction. The use of 3D technologies and 3D interaction techniques allow creating new interfaces which provide more natural and intuitive interactive software.

In this thesis we propose novel methods for 3D interaction in 3D environments. The evaluation of these methods was performed based on three interaction environments: 3D interaction using portable multi-touch devices, 3D hand gesture data manipulation using 3D database representation and 3D multi-threaded programming using hand gesture interaction. The three experiments provided qualitative and quantitative information necessary to evaluate the features of the presented interfaces.

The first experiment, based on the use on the use of portable multi-touch devices, was seeking to evaluate the use of 3D movements to interact under a 3D environment. Also the possibility of generating collaborative interaction under 3D interfacing (simulating a 3D multi-touch table top environment) was evaluated.

The second experiment consisted of 3D touchless data manipulation, removing the intermediate device (portable multi-touch) and providing hand gesture data interaction using the Kinect device. Furthermore, this evaluation was conducted over a 3D cube database model, based on the concepts of multidimensional databases and graphic databases.

The third experiment intended to evaluate the possibility of software generation using a 3D interaction environment, following a similar model of interaction from the second experiment, but providing better two handed interaction. The environment addressed multi-threaded programming under a 3D interface.

The three experiments provided valuable data about users' interaction and preference, which were tested with users of different ages and levels of knowledge. The research process and results are summarised in this research.

Declaration

I hereby declare that this thesis entitled "Advanced Computer Vision-Based Human Computer Interaction for Entertainment and Software Development" is the result of my own research except as cited in the references. This thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Parts of this thesis have been published in the following:

[1] Raul A. Herrera-Acuña, Vasileios Argyriou, Sergio A. Velastin, "Toward a 3D Hand Gesture Multi-threaded Programming Environment", in Advances in Visual Informatics, , Proceedings Third International Visual Informatics Conference, IVIC 2013, Selangor, Malaysia, November 13-15, pp. 1-12. (2013).

[2] R. Herrera-Acuña, V. Argyriou, S. Velastin, "Graphical interfaces for development exploiting the third dimension using Kinect", Proceedings of the 9th International Conference on Intelligent Environments, vol 17, pp. 356-367 (2013).

[3] Rob Dupre, Raul A Herrera Acuña, Vasileios Argyriou and Sergio Velastin: "3D Interaction environment for free view point TV and games using multiple tablet computers", BAGS Workshop, CVPR, June 23-27, Oregon, USA (2013).

[4] RA Herrera-Acuña, V. Argyriou, S.A. Velastin, "Portable Multi Touch TableTop: a composite approach for industrial applications", 2013 Constantinides International Workshop on Signal Processing, IET, Jan 25, Imperial College, London. (2013).

[5] R. Herrera-Acuña, V. Argyriou, S.A. Velastin, "The Evolution of Multi Touch Table-top Systems", Latin American Conference on Networked Electronic Media (LACNEM), Chile, 2012.

[6] R.A.H. Acuña, C. Fidas, V. Argyriou and S.A. Velastin, '"Toward a Two-Handed Gesture-Based Visual 3D Interactive Object-Oriented Environment for Software Development," Proceedings of the 8th International Conference on Intelligent Environments (IE), pp. 359-362. (2012).

Papers under review:

[7] R. Herrera-Acuña, V. Argyriou and S. Velastin "A Kinect-based 3D hand-gesture interface for 3D databases", Journal on Multimodal User Interfaces, 2013-2014.

[8] K. Jablonski, R. Herrera-Acuña, V. Argyriou and S. Velastin "Cooperative 3D Interaction framework for group applications using Augmented Reality", International Journal of Computer Science and Artificial Intelligence (IJCSAI), 2014.

[9] R. Herrera-Acuña, V. Argyriou and S. Velastin "A software development interface based on 3D interactions and 3D metaphors" International Journal of Human Computer Studies, 2014.

Acknowledgement

I would like to thank my supervisor Vasileios Argyriou for his invaluable support and guidance throughout my PhD. His continuous hard work made possible the submission of this thesis

Also, I would like to thank my second supervisor Sergio Velastin for his support, comments, critical view of my work and especially for his friendship.

Special thanks to all my friends and colleagues at the Digital Image Research Centre for their continuous support. Moreover, I would like to thank to all my friends that have been there giving me strength, even when I did not believed in myself.

I am grateful to the Universidad de Tarapacá, especially to the Computing Engineering Area, which have been sponsored my PhD.

Finally, I would like to thank my family, and specially my parents for their love. Even when they are far away, their love and support had been fundamental to all my achievements during this adventure.

Raul Herrera Acuña

Chapter 1

Introduction

1.1. Context and Overview

Human Computer Interaction (HCI) can be defined as "*the study of the way on which computer technology influences human work and activities*" (Dix, 2009). Given the previous definition implies that HCI lies at the crossroads of many scientific areas including artificial intelligence, computer vision, gesture recognition and motion tracking. In recent years there has been growing interest in improving all aspects of interaction between humans and computers. It is argued that to truly achieve effective human-computer intelligent interaction (HCII), there is a need for the user to be able to interact naturally with the computer similar to the way human-to-human interaction takes place (Huang et al., 2011; Lokesh et al., 2010). New approaches to computer-human interfaces (Jaimes and Sebe, 2007) like multi-touch systems, luminous rooms, gesture interpretation devices and tangible user interfaces created a new need in the software design and development area: systems able to generate these applications using the same metaphor.

The design of multi-modal systems has been a goal for many researchers, especially for those who work in the human-computer integration area. Furthermore, the need for new systems to replace conventional interface devices is becoming an important issue for scientists. Multi-modal systems provide a new natural way to interact with computers and access relevant information based on the rules of each working environment keeping the focus on the interaction with that information, as natural as possible.

Advances in hardware and software integration have created new ways to interact with machines. These advances are aiming to improve human-computer interfaces with their main objective being to become more natural, based on body motion understanding, gestures and sensory integration plus being able to understand beyond just written commands (Blackwell, 2000). This effort is pioneering a new era in the design and development of systems that is more suitable to operate in specific areas like geographic data management, education, industrial design, architecture, databases and web interfaces. The area that is leading the advances in interaction is the video game industry due to the need to provide new levels of experience and much higher interaction between users and the systems. In addition most of the video game interaction systems have been used in scientific areas (i.e. as graphics, tracking, body motion analysis, etc.), mainly as new hardware devices for acquisition making that technology universally available.

Contemporary graphical interfaces have evolved from the typical console-based ones to visual 3D environments (Dang et al., 2009) where the user can interact with components but there are still limitations in this interaction. Significant problems with 3D interaction and manipulation of objects remain. The poor use of depth in the systems and combining them combine it properly with typical 3D interfaces (Teyseyre et al, 2009) are open research areas. The interaction is not yet adaptable to the needs of users and is restricted to a few limited interactive commands.

The connection between the graphic metaphor and the data being manipulated (Wu et al, 2003) can be problematic and is highly dependent on the context, limiting the creation of reusable components. Consider the case of a 3D software design and development framework where the developer has to create the software layers for the specific components, including the connection of the graphic elements with data and the interaction language (Fishkin, 2004). Furthermore the developer has to design and create the objects that are going to be used as basic design and programming elements, flexible enough to develop new applications (Conway et al., 2000). Designing flexible components and interaction mechanisms to be used in 3D environments for any kind of interface is an open problem thus software design and development needs to be improved incorporating new ways of interacting with the system.

1.2. Aims and Objectives

The aim of this project is the design and development of a novel architectural framework in order to create new software using a natural interface based on multi-touch devices and hand gestures whilst trying to overcome some of the most problematic issues in the area of HCI.

This work is focused on the conceptual and architectural design of a multi-layered framework aiming to provide both multi-touch and two handed gesture-based 3D environments for software development and entertainment. In this context, the main objectives of this project are:

O1.To define a multi-user interactive 3D framework, using portable multiple point of contact interaction devices (tablet Personal Computers –PCs- or mobiles) generating collaborative and fused 2D and 3D table-top interaction mechanics. This objective aim to problem of unnatural interfaces that do not allow direct collaborative interaction in working environments.

O2.To introduce a novel two-handed gesture based 3D environment for software development including interaction styles, architectural design and an adaptable interface to the application. This objective pretends to solve the problem related with the manipulation of graphic components, making more natural the interaction with software (based on the use of the hands, as the human beings use to do it in the real world) and solving the problem of creating reusable graphic components.

O3.To define an evaluation methodology for 3D hand gesture based interfaces using both qualitative and quantitative measurements. This objective has been defined to assess the advantages of the previously presented to assess how beneficial are compared with traditional approaches.

The activities to achieve the research objectives described above include:

- Research on interaction mechanisms for software development and entertainment using multi-touch devices and their internal hardware features both in 2D and 3D.
- Establishment of layer architecture for gesture-based frameworks
- Definition of experiments to assess 3D graphical environments and evaluate user experience.
- Definition and experimental evaluation of a programming 3D hand gesture based framework for multi-core software development.
- Collection of experimental data from a population of subjects during the use of the novel 3D frameworks.

These research objectives will generate a framework capable of improving the process of software development, reduce learning time and will increase the use of natural 3D interfaces in Human-Computer Interaction (HCI) mechanisms.

1.3. Applications of HCI

Since the beginning of computer sciences interaction with machines has been an important research area that has evolved from complicated console code based interfaces to gesture based interaction in less than half a century. Advances in several related areas such as computer vision, computer graphics, tracking, cognitive science and hardware development have all contributed to its evolution (Michalski et al., 2008).

User friendly interfaces increase productivity and improve data visualisation allowing complex metaphors to be represented using multimodal interfaces are technological advances that have improved the HCI. These improvements introduced several new communication mechanisms and novel types of interfaces (Jaimes et al., 2007).

There are several application areas in which advanced interfaces and more friendly environments are essential to human computer interaction. In the entertainment industry, and especially in video gaming interaction hardware has improved significantly, re-shaping the vision about what and how human computer interaction should be (Marsh, 2011). The new hardware generation (e.g. Kinect and Wii) is a technological advance capable of capturing movements in real time reflecting the actual behaviour of the user in the game. In a similar manner multi-touch devices that have been popularized by portable game systems allow the user to interact with the game reflecting his actions in real time, allowing to create new interfaces with computers. Other entertainment areas have made use of HCI to improve the user experience such as within television and movie industry, where the use of 3D data capturing and visualisation is leading to new and more interactive content development (Lino et al, 2010; Ronfard and Taubin, 2010; Francese et al., 2012).

Another area where advanced HCI techniques have improved user productivity is bioengineering. One example is the technological advances in modelling and manipulation of proteins where advances in 3D interaction and geometrical modelling allows users to interact, connect and even create their own protein representations from scratch, and at the same time to visualise the results (Crivelli et al., 2004). Another area that has been benefitted from advances in HCI is design, moving from complex industrial machinery to home devices. The use of 3D interaction has boosted the capabilities and performance of design teams in turn reducing production costs and minimising product failures (Velenis et al., 2009). Robot interaction and programming enhanced their results by incorporating 3D interactive HCI. The use of new interaction hardware and actuation interfaces in the last twenty years have sparked a revolution in robots' interaction with their environment (Kanda et al., 2002) and changed the way they can be operated and monitored (Micire et al., 2012).

It can be argued that 'information systems' is the area that was mostly influenced by advances in HCI. In the beginning, vast amount of data were stored with a lack of context and so required significant expertise in this area to use the available information in an efficient way (Stefanidis et al., 2011). Databases reduced that problem and later HCI improved the context visualisation by adding multiple types of graphical representations of data objects but was limited to two dimensions. A more compact representation of data in multidimensional databases can be obtained by introducing 3D interfaces contextualizing and interconnecting information that was previously considered to be unrelated (Vasilliadis, 1998). Multidimensional databases can represent data in more than two dimensions and can correlate large amounts of information that can be visualised and manipulated using 3D techniques based on touch or gesture recognition.

Software development can be improved by the use of 3D environments, allowing multiple views of the working elements and detection of errors that will not be evident in a traditional 2D development environment. Moreover, the possibility of a three dimensional view allows parallel developing in an easier and more user-friendly way especially for high performance graphic applications.

1.4. Review of related hardware technologies

Interaction interfaces with computers have changed drastically during the last few decades. Initially the only way to interact with computers was by using complex hardware based on switches without any graphic feedback, but with paper tape. Later, the development of bi-colour screens allowed the use of direct input using keyboards that permitted the creation of the first text-based graphic interfaces. These text-based interfaces were the first to provide direct feedback to the user. Advances in other technologies such as graphic cards, processors and the advent of the mouse allowed the creation of more complex and advance graphic interfaces where the text was just a part of the software and other structures, such as diagrams, graphic illustrations of objects and 3D representations of concepts became an important part of the interface. The use of 3D representation allows the user to have a more natural and real-world related interaction with object, especially beneficial on interfaces where simulations of the real world are necessary. A few years ago, other models of interaction appeared such as touch displays, gesture-based interaction, virtual reality environments and even direct human brain-computer interfaces that further change the way we interact with computers (Sears and Jacko, 2007). As a result, the hardware evolution is directly linked to the evolution of software interfaces;

therefore it is necessary to present an overview of the hardware evolution along the years.

Nowadays the development of new hardware technologies allows the creation of software based on new interaction paradigms such as multi-touch interaction and gesture-based interaction. Therefore an overview of hardware evolution related to HCI is essential to understand these advances over time. Not just software interfaces but also the interaction between users and computers were directly associated to the use of external devices such as keyboards or mice and the results of that interaction were displayed on a screen (North et al., 2009). The traditional paradigm "windows, icon, menu, pointing" (WIMP) has evolved slowly to more natural and free interfaces that provide a more efficient interaction.

The advances in the last 40 years have addressed those needs and two specific areas have presented a radical evolution: multi-touch and hand gesture interaction.

1.4.1. Multi-Touch Table-tops

The multi touch term refers to devices capable of retrieving position information of several contact points on a touch sensitive surface. Generally these systems provide direct feedback about the interaction in a separate screen or, as it is typical nowadays, directly on the contact surfaces that act also as displays. The multi touch table-top term refers to large displays, generally not portable, where multiple users can interact with the surface and have direct and personalised feedback, depending on the software and the device.

The multi touch systems have become the most appropriate alternative to substitute the traditional human-computer interfaces (based on the use of mice and keyboards, with a separated display screen) because they allow the user to interact directly with graphic elements on screens (Kin et al., 2009). During the last decade this technology evolved gradually until it reached a level advanced enough to be integrated into common devices such as mobile phones, video game consoles, laptops, etc. (Malik, 2007).

Multi touch technology came about in 1982, when Nimish Mehta, researcher of the University of Toronto, developed the "Flexible Machine Interface" currently considered the first multi touch system (Saffer, 2008). "The Flexible Machine Interface" could support multiple and simultaneous contact points, allowing the user to perform graphical manipulations. Later Bell Labs tried to continue this work publishing and developing more advanced devices. This field of research was not recognised widely until 1985, when Bill Buxton in collaboration with the "Input Research Group" of the University of Toronto developed the "Multi-Touch Tablet" (Mazalek, 2005). The "Multi-Touch Tablet" device (see Figure 1.1) was able to detect an arbitrary number of inputs produced by multiple simultaneous touches performed by the user on a special surface. The device was able to identify the touch's coordinates and estimate the degree of pressure in each contact point.

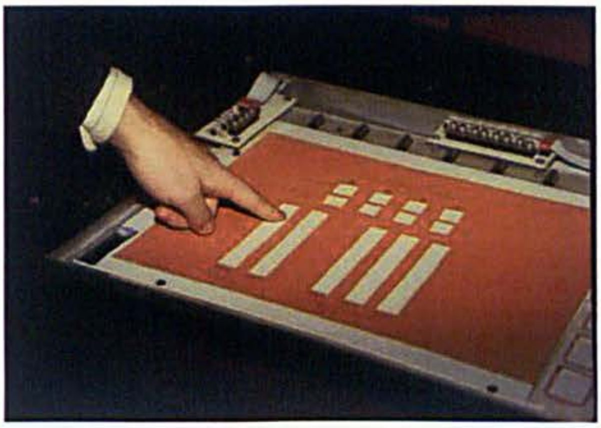


Figure 1.1: First Multi-Touch Tablet (Buxton, 1985).

During the last few years, different technologies have been developed in the field of multi touch surfaces, improving the interaction capabilities based on contact devices, such as pens, gloves, cubes, magnifying glasses, cards, etc., or systems where the interaction is provided by bare hands.

Later, devices based on infra-red light replaced electronic systems and in 1997, Matsushita and Rekimoto published their work on HoloWall (see Figure 1.2), which is a computer wall that allows the user to interact with graphic objects without the need of any external electronic devices. The user is able to interact using fingers, hands, body or even objects.

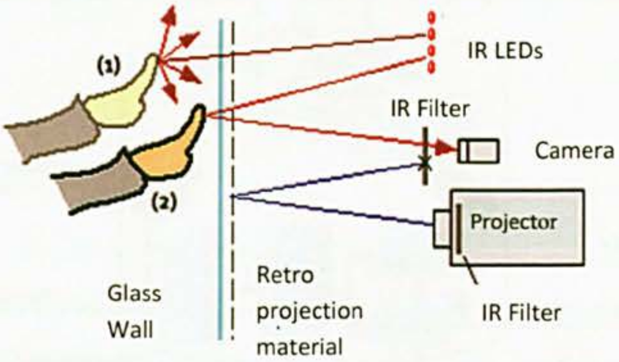


Figure 1.2: System configuration of multi-touch HoloWall (Matsushita and Rekimoto, 1997).

In 2001, Mitsubishi Electrical Research Laboratory (MERL) launched the DiamondTouch (see Figure 1.3) a touch sensitive table created for enhanced collaborative work between several users providing visual contact with the information. A representation of interaction with the device can be seen in Figure 1.3.

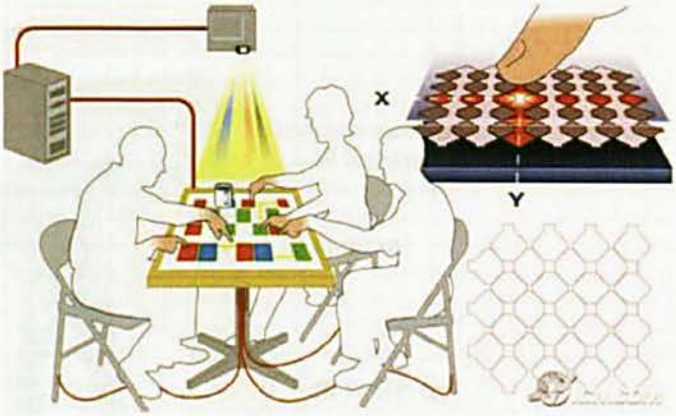


Figure 1.3: Interaction representation for DiamondTouch system (Dietz and Leigh, 2001).

Multi-touch systems have been used in artistic areas too. ReacTable was designed and developed in 2003 at University Pompeu Fabra in Barcelona. ReacTable is an electronic collaborative musical instrument with a tangible interface, based on a multi-touch table, capable of identifying and following fiducial markers, allowing the control and combination of several musical instruments. A schematic representation of this device appears in Figure 1.4.

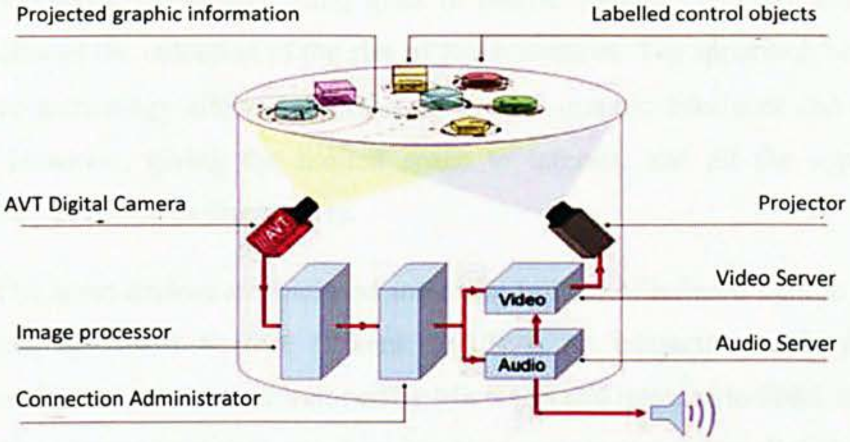


Figure 1.4: Multi-touch ReacTable system's configuration (Jorda et al., 2005).

The devices also started to integrate 3D displays. For example, TouchLight (Wilson, 2004) is a touch screen for visualisation and interaction with 3D images. It was developed by Andrew D. Wilson from Microsoft and presented at the end of 2005. An example representation of this device is shown in Figure 1.5.

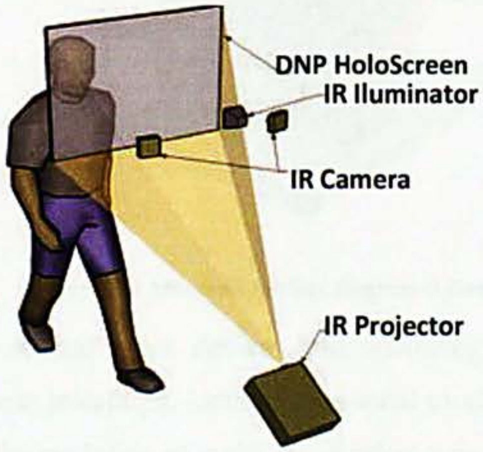


Figure 1.5: Touchlight working scheme.

The apparition of I-Phone on 2007, one of the first mobile to use a compact electronic surface capable to detect multi touch interaction, allowed new advances on touch interaction. The use of the Pro-Capacitive touch technology, generating an electronic capacitive surface, using glass or plastic materials (Barrett and Omote, 2010), allowed the reduction of the size of touch surfaces. The apparition of the pro-capacitive technology allowed the creation of new graphic interfaces and software related. However, giving the limited space to interact, and all the applications developed just aimed to single users.

The latest devices are enclosed and make full use of infrared light to generate interaction. Microsoft Surface (Likens, 2010) is an interactive table that uses navigation by tactile menus, developed by Microsoft and released in 2008. Microsoft Surface has a transparent surface where images are projected from a digital projector placed inside a table. Additionally, four infrared cameras are placed at its corners and one at the centre of the table to detect the infrared occlusion on the surface, caused by user's activities. A schematic view of this device is presented in Figure 1.6, showing (1) the interaction surface, (2) the infrared emitters, (3) the infrared cameras and (4) the digital projector.

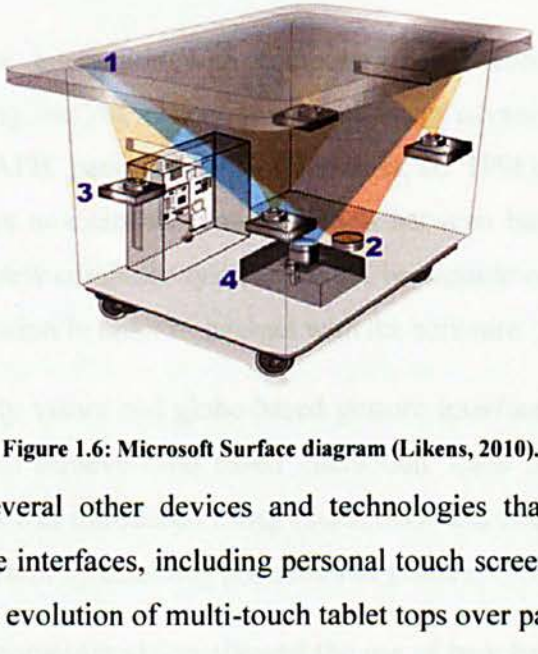


Figure 1.6: Microsoft Surface diagram (Likens, 2010).

There are several other devices and technologies that contributed in the development of these interfaces, including personal touch screens and smart phones. Figure 1.7 shows the evolution of multi-touch tablet tops over past decades.

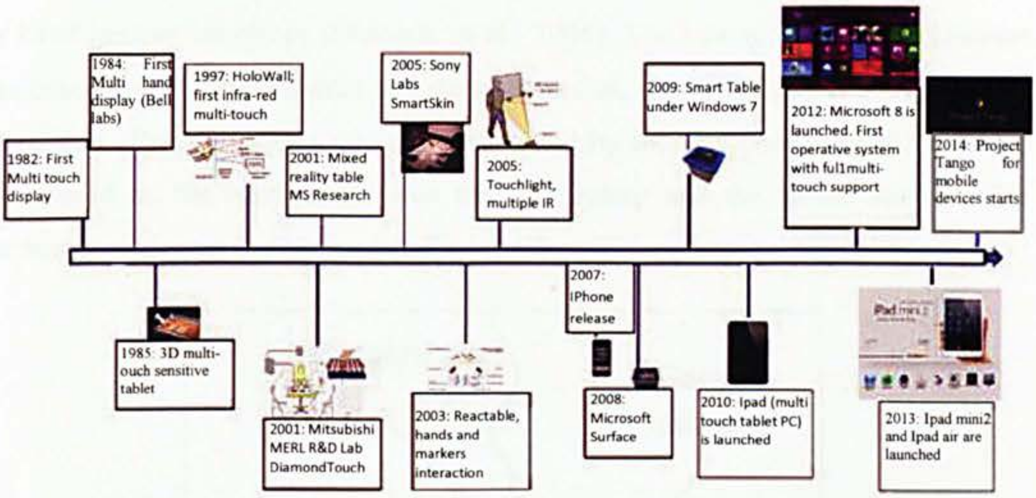


Figure 1.7: Multi-touch table-tops evolution.

1.4.2. Hand gesture based interaction

Research to improve interaction with computers has become one of the main research issues during the last twenty years and many advances have been made. During the SIGGRAPH panel in 1998 (Harris et al, 1998), the importance of developing new ways to establish communication between humans and computers was addressed. The new computer systems should be capable of capturing all forms of human communication in order to interact with the software.

Virtual Reality visors and globe-based gesture interfaces were a few of the initial devices used to achieve hand based interaction. Later the CyberGlove from Virtual Technologies was introduced using neural networks and three dimensions to operate the whole system by detecting postures and gestures (Nishino et al., 1997).

Advances in computer vision allowed the use of bare hands under single and multiple camera systems to interact with 3D environments. Simple gestures such as movements and pinches were used to manipulate objects. These kinds of systems used to work with a similar metaphor to the one used in multi-touch systems, where just the finger tips are detected and the movements on 3D were limited to the intensity of the tip detection (Segen and Kumar, 1998).

The evolution of the acquisition devices also permitted skin colour detection for hand gesture interfaces (Dhawale et al., 2006). The interaction in these systems was based on hand movements and shape detection, allowing open and closed hand interaction. The 3D movements were estimated by the size change of the detected hand based on the contrast between the skin colour and the "desk" surface. The interaction desk area is displayed in Figure 1.8.

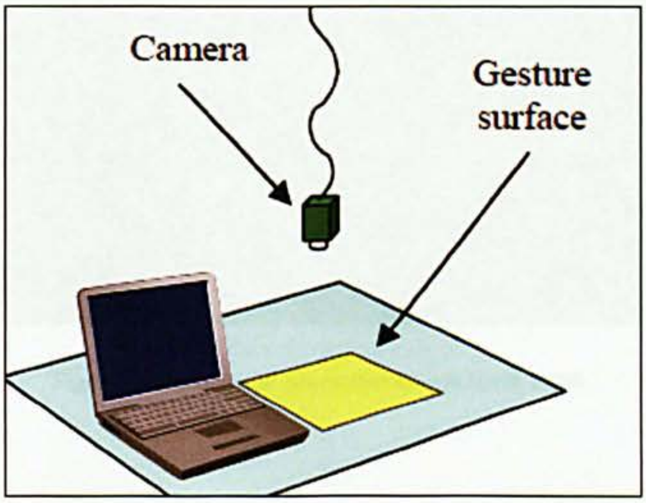


Figure 1.8: Hand gesture colour and size/shape based interaction desk (Dhawale et al., 2006). The gesture surface provides an area of interaction, where the hand is identified by colour segmentation.

Improvements in cameras, displays and the use of lasers to detect depths, shapes and hands with higher accuracy led to the next generation of interfaces usually called "tangible". One example is the work of Ratti in the MIT's Tangible Media Group (Ratti et al., 2010) that presents an alternative replacement for the text-based systems in geographic information systems (GIS). This new approach permits interaction with geographical data where the user can modify the interface using tangible objects, (such as blocks, trees, hills, etc.), integrated with augmented reality (see Figure 1.9).

Figure 1.9: SandScape interaction system (Ishii, 2008).

Nowadays, advances in video game technology have contributed to the development of new devices for human computer interaction such as Microsoft Kinect. These systems use depth, shape and segmentation techniques to determinate the hands' locations and subsequently the fingers using hand shape detection, convexity calculation operations to detect individual fingers and machine learning, (Van den Bergh et al., 2011). Kinect's system architecture is shown in Figure 1.10.

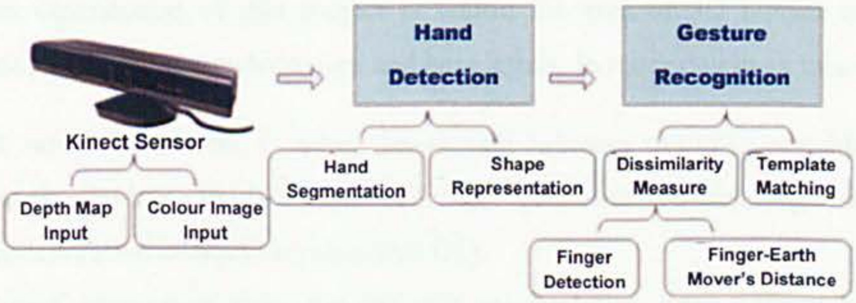


Figure 1.10: Typical Kinect based hand gesture system architecture (Ren et al., 2011).

Compared with multi-touch technologies, hand-gesture interactions seem to progress slower but the capabilities of 3D interaction systems are numerous (Jackson et al., 2012).

The next image summarises the evolution of hand-gestured based interaction environments and the related advances in hardware.

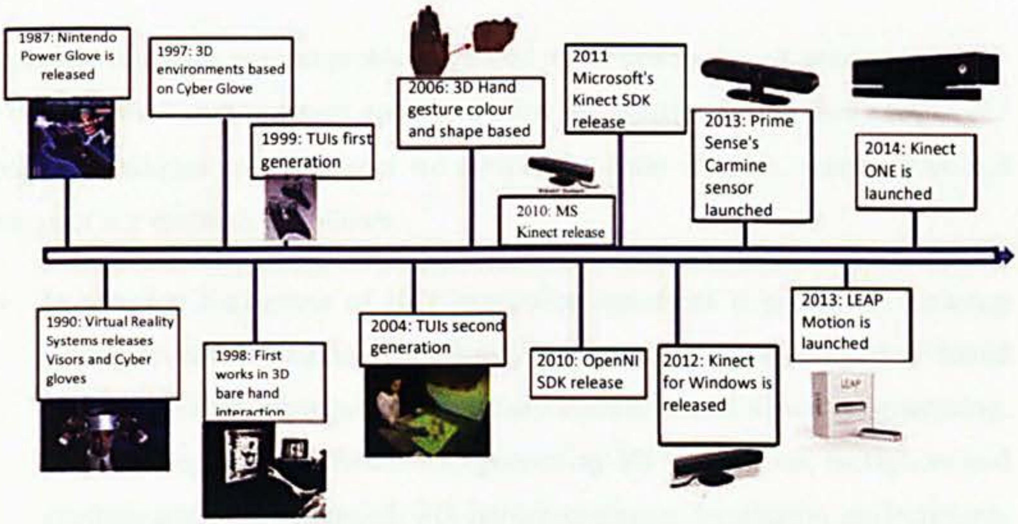


Figure 1.11: 3D Hand gesture-based interaction and evolution.

1.5. Contributions

The main contribution of this project is within the area of 3D human computer interaction, using multi-touch devices and bare hands. In more details in this study:

C1. A novel framework to allow interaction between multiple portable multi-touch devices in 2D and 3D was proposed considering multi-user collaborative interaction (objective O1).

C2. Novel interaction styles for 3D data manipulation were defined based on hand gestures that try to improve user interfaces and simplify the overall interaction procedures in 3D datasets (objectives O2, O3).

C3. A novel framework for multithread programming in a 3D environment was suggested improving the code efficiency, the overall robustness of developed software and reducing development and learning time (objectives O2, O3).

1.6. Structure of Thesis

This project considers several problems related to 3D gesture based interfaces, multi-user collaborative environments and multi-view information fusion. For each of the proposed techniques and interfaces we outline the main features, assumptions and offerings of our methods as follows:

- In **chapter 2** a *review* of HCI interaction interfaces is presented, focusing mainly on new interaction paradigms, such as multi-touch and gesture-based interaction, data management in 3D environments and visual programming. The challenges and difficulties in generating 3D interactions, metaphors and environments are discussed. 3D human-computer interaction environments are analysed and related datasets and metrics used to evaluate performance and quality of these systems are presented. Finally, a review of evaluation methodologies for user interfaces is performed, including evaluation methodologies for 3D interfaces indicating challenges, difficulties to overcome and future research areas.
- In **chapter 3** a novel framework for *multi-touch 3D object interaction* systems is presented. Important aspects are explored including interaction techniques, interface features and components involved in the process of multiple 3D interactions. A full description of the proposed method is performed, including a step by step definition of the contributory interaction styles. Finally an analysis of the experimental results is provided including a discussion on the issues related with the device error and its propagation. The experimental work in this chapter was performed with the collaboration of Rob Dupré, in the research related with the inertial devices, and Konrad Jablonski, in the implementation of the 3D and 2D interface to evaluate the methodology presented.

- In **chapter 4,** the approach proposed for *3D hand-gesture based HCI and data interaction* is summarised, and a novel database programming interface presented. The differing interaction styles and available functionalities are fully explained, including a step by step analysis of the interaction process and visualization elements. An evaluation of this interface is presented describing the experiments plus providing comments on the outcomes. Finally conclusions are presented and future work is discussed.
- In **chapter 5** a novel *3D hand-gesture based programming environment* is suggested, explicitly defined for multi-thread programming. A complete definition of interface involved graphic elements; and interaction techniques are presented. Novel features and interaction scenarios are presented comparing this approach against traditional programming interfaces. Also a set of experiments to validate the performance of the proposed system that includes a full description of the methodology, the evaluation procedure and the obtained results are discussed. Finally in this chapter, conclusions and comments on possible related applications are presented.
- In **chapter 6** we summarise the main contributions and offerings of this study and discuss possible directions of future research.

Chapter 2

Literature Review

Studies concerned with interaction between humans and computers have been a prolific research area since the birth of computer science. The main interest area of these studies concentrates on enabling the definition and creation of interfaces easier to use and understand, thus continuously improving the way users interact with computers. Original interactions with computers based on punch cards, paper tape and line editors have evolved quickly to graphic interfaces, passing through several technologies that now are obsolete (Sears and Jacko, 2007). However advances in HCI are still afoot and aim towards ever more world-like interactions. Progress on interfaces that resemble human interaction with the real world, (usually known as "natural interfaces") is becoming more common (Wigdor and Wixon, 2011), allowing the creation of new interface technologies for multi-touch applications, 3D data interaction and visual programming.

Given the three dimensional nature of the real world, the use of 3D technologies to represent concepts and interact with them is an open and interesting research area. Several studies related with the use of 3D interfaces for HCI compared with 2D interfaces have proved that the use of 3D elements provide advantages such as better understanding of concepts and memorization (Cockburn 2004), reduced time of training in the use of graphic menus (Kim et al., 2011), reusability of components (Xavier et al., 2008) and the possibility of generate more natural and immersive interfaces (Chen, 2006). Contemporary graphical interfaces have evolved from the typical console-based writing code to visual environments where the user can interact with components to create new applications. However current human-computer interaction systems still have non-graphical elements that could be replaced by iconic representations. Moreover, human-computer interfaces and the conceptual representations of interactive graphic components can be improved with a 3D graphical user interface. Some aspects of data management such as representation of concepts, interaction with complex associations of information and discovering hidden relationships between information sources, could be easier to understand if the metaphor is part of the representation in a 3D environment (Chittaro et al, 2009).

Thus it's desirable to fully move towards 3D based interactive systems. This need has triggered significant advances in software and hardware development by combining technologies to generate new interfaces.

Improvements in interactive multi-touch mobile/table-top systems and hand gesture interaction technologies have promising applications in the area of human computer interaction and 3D interfaces.

In this chapter we review HCI interfaces based on tangible devices and hand gesture mechanisms for cooperative applications, database interaction and software development systems. We also explore the evolution of multi-touch interfaces and review related interaction techniques and extensions for the third dimension, compared to traditional interfaces. Furthermore we analyse the two-handed interaction research progress and how the hardware evolution have allowed interaction methods to improve their performance. We will then discuss the challenges and benefits of 3D data interaction both from users and developers perspectives, presented in applications related to databases and software development. Finally, different evaluation techniques for interactive interfaces are presented including evaluation procedures for 3D based systems.

2.1. Multi touch systems

The need for more intuitive interfaces became apparent when the first "Window, Icon, Menu, Pointing" (WIMP) devices were introduced in 1980 (Volk et al., 2011). More intuitive interfaces has become a matter of urgency during the last few decades particularly due to the advances in ubiquitous computing and the consequent appearance of the Web 2.0, where the use of multiple interactive graphic elements require more than a single point of interaction and text based interfaces (Mika and Greaves, 2012). These two aspects made apparent the need for interfaces and devices able to interact with multiple elements simultaneously, which is the key aspect of multi-touch devices (Calentano and Minuto, 2008).

The technology used to capture interaction in multi touch devices can be categorised in three types: devices based on hardware, software and hybrids. Devices based on pressure sensitive hardware, where the surface electronically identifies the position of the pressure point was the first approach to obtain real time multi-touch feedback. This technology can be found in devices with small and limited contact area, such as Smart Phones, Tablet PCs and DiamondTouch table (Dietz and Leigh, 2001), where the number of contact points is restricted, and also corresponds to the first devices of this kind (Mehta, 1982) . The second category corresponds to multi-touch devices based on computational vision and infrared image capturing, where the interaction point (generally related to the finger tips) is captured by cameras (the interaction method can be seen in Figure 2.1). In this case, segmentation techniques are used to obtain the contact points on a defined interaction area. This technology is used in the majority of the table-top displays, where the amount of contact points and users is limited only by the size of the device. The use of infrared light-emitter diodes (LEDS), cameras capable of capturing infra-red light and contact surfaces with projection materials has led to the creation of larger interactive surfaces (Figure 2.2). Later, these systems evolved to enclosed devices, where the problems of light interference have been almost eliminated. The use of multiple cameras and light manipulation techniques, such as total internal reflection of infrared light in the interaction surface (using large arrangements of infra-red LEDS and lasers in some occasions), were the elements that required less expensive, larger and durable devices. These advances have increased research in this area and on these technologies (Chang et al., 2010, Kim et al., 2007). However these techniques are not able to identify different users and simply provide general interaction feedback without distinguishing who interacted or how the user did it (there is not a specific detection of hand or finger during the interaction). Currently these interfaces are solely available for large vision-based multi-touch surfaces, without the option to be used for tablet pcs in a collaborative way.

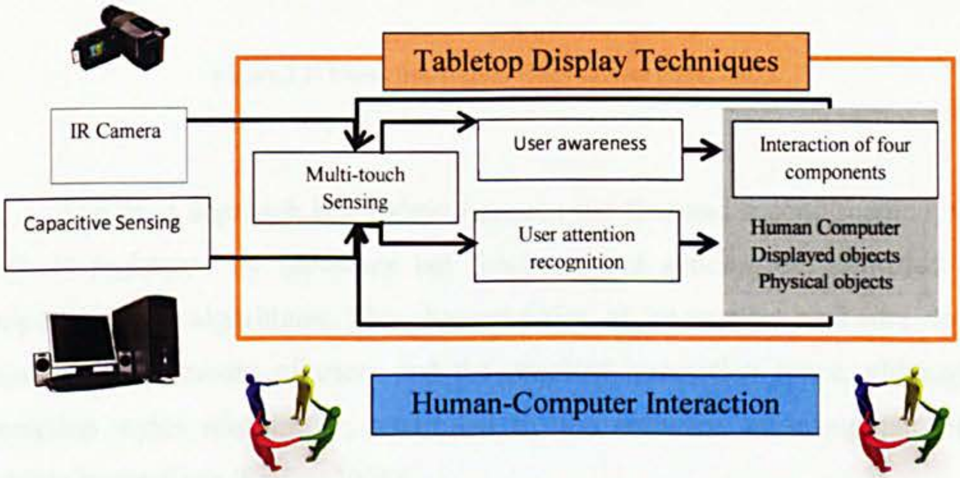


Figure 2.1: Multi touch table-top human-computer interaction (Kim et al., 2007).

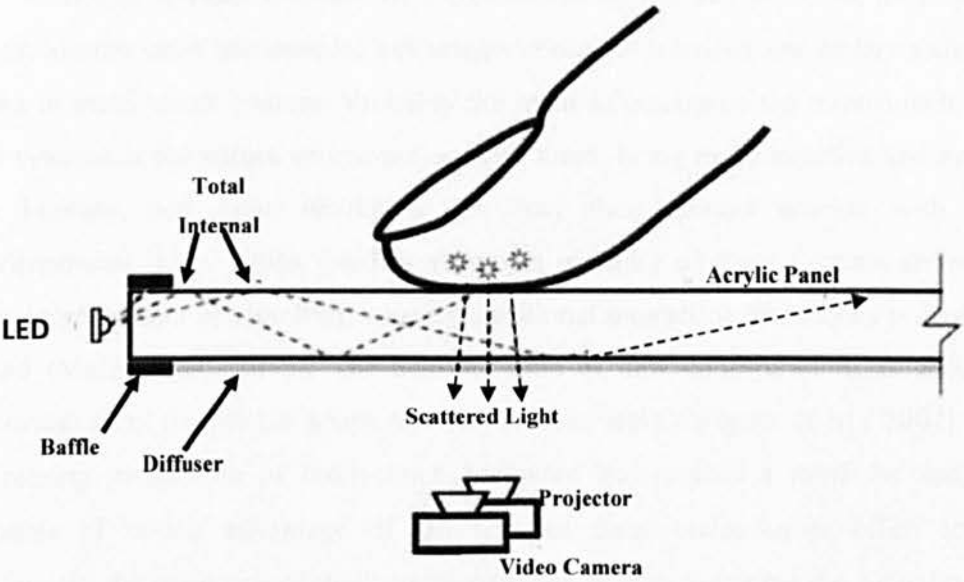


Figure 2.2: Vision based multi touch surface (Han, 2005).

The third approach is a hybrid between the first and second method, where touch is registered by hardware but feedback and actions are identified using computer vision algorithms. The characteristics of interaction hardware depends mainly on the amount of users and the required interaction space, although the interaction styles are similar, controlled by the software, allowing parallel and multiple interactions (Saffer, 2008).

One of the main factors for the increase in research on multi-touch interfaces for table tops is due to the multiple benefits provided by this technology. From its conception several advantages over conventional interaction devices have been observed (Buxton, 2010). The main benefit of these devices is their capability of being "self-contained", (no need for external interaction devices), allowing direct interaction between the user and the objects visualised on the screen. Another advantage is the limitless number of interaction points that enables users to perform multiple actions and manipulation tasks with the graphic elements, which is the main deficiency in traditional interaction devices that just offer one point of interaction at any given time (traditionally via keyboard or mouse). Due to their solid construction and lack of external components, multi-touch table tops tend to be more durable in environments of constant use.

The advantages provided by these devices are not just limited to the physical layer. Similar work has unveiled advantages related to usability and understanding of tasks in multi touch systems. Probably the main advantage of the multi-touch table top systems is the nature of interaction with them, being more intuitive and natural for humans, and more similar to the way that humans interact with their environments (Han, 2006). Studies about the usability of these systems show that they increase user productivity, causing traditional interaction techniques to look out dated (Malik et al., 2005). The learning time of new configurations is speedier, allowing users to quickly adapt to new environments (Wigdor et al., 2007). The increasing production of multi-touch hardware has created a need for software capable of taking advantage of the features these technologies offer. In the beginning, the main aim of multi-touch software was to overcome the limitations of traditional input interfaces (i.e. keyboards, mice, trackballs, etc.) by implementing a new visual interaction vocabulary based on 2D gestures over the touch surface. These gesture based interfaces were able to recognise all the typical keyboard and mouse (2-button and 3-button) operations including a set of gestures based on natural finger movements whilst also having the option to create new sets of gestures, (Westerman and Elias, 2001). These first studies highlighted advantages over traditional interaction specifically in the area of learning, since multi-touch interfaces speed up task performance times. It can be noted though that technical limitations of touch detection in the first commercial devices limited more advanced interface development hence supporting only simple contact interfaces. Key to the future of these technologies is related to the possibility of collaborative group work. This will be applicable not only to professionals in this field, but also among professionals of different subject matters. This is true particularly in relation to interactive design applications where knowledge from different perspectives can be beneficial (Clifton et al., 2010). These types of devices can generate interaction between professionals and the general public because of the intuitive interaction mechanics providing new applications in different areas. A clear example is education whereby the use of multi-touch technology as a collaborative tool for students and teachers can effectively replace typical single-point interaction methods (Cheng et al., 2009). In medical education, multi-touch table tops can be used to provide virtual interaction with a patient (Figure 2.3) reducing potential risks and costs (Kaschny et al., 2010). It can also help to implement new educational models, such as object-based learning for children (George et al., 2011). Combination of multi-touch and augmented reality to improve the learning time is also another area of research with positive results (Jang et al., 2007). Multi-touch tables are said to improve social relationships in communities with applications in quality of living, such as environmental design for neighbourhoods and buildings (Fernquist et al., 2010).

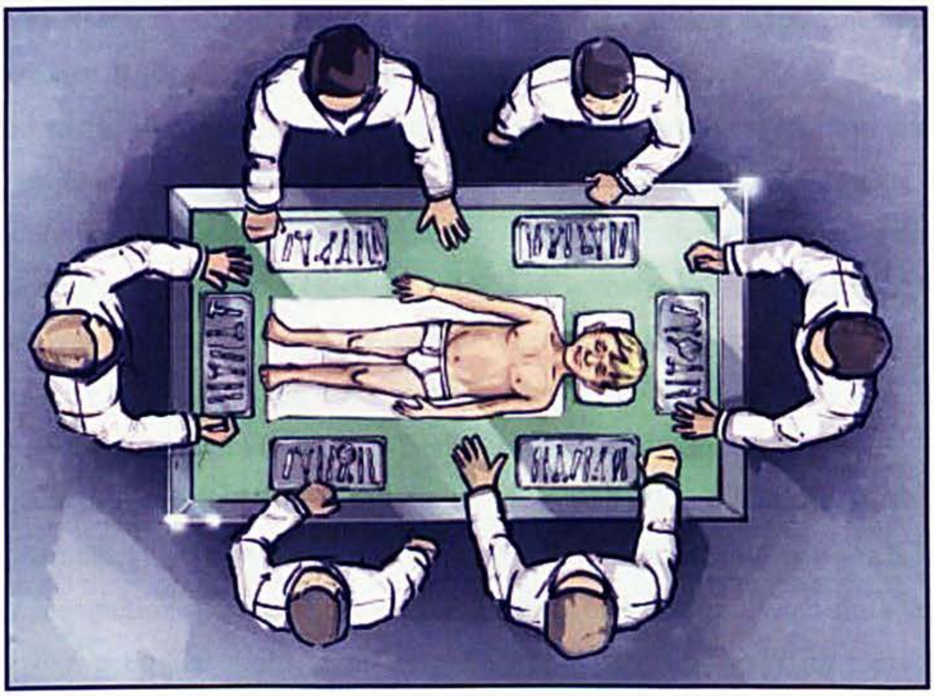


Figure 2.3: Interaction on virtual patient (Kaschny et al., 2010).

Research to compare multi-touch interfaces with other interfaces is based on traditional devices such as mice and working environments, where two or more users can interact directly with the touch surface (collaborative environments). Experiments to compare multi-touch table top interfaces versus multi-mouse Single Display Groupware (SDG) setups (Hansen and Hourcade, 2010) enabled the identification of those activities deemed more suitable for multi-touch interfaces and traditional input mechanisms (e.g., single point interaction). The results suggest that despite some efficiency issues, multi-touch collaborative systems (systems where two or more users can interact at the same time and collaborate in order to achieve a specific task) provide several advantages in the interaction and learning time of new interfaces, making touch interfacing preferable to mouse-based systems, especially when the users work in pairs. However, the study previously mentioned does not consider the advantages of using portable multi-touch devices and all the internal systems (such as accelerometers, Global Position Systems –GPS- and gyroscopes) that provide spatial awareness of these devices in a collaborative environment.

In mobile devices there have been several studies related to the use of touch features. Research has been performed to acknowledge how important the touch performance is for users (Rukzio et al., 2009), where factors such as data security (related to the process of inserting data to the device), speed and intuitiveness have been evaluated. Considering the previously mentioned factors, physical touch interaction shows the highest preference by users over other interaction techniques provided by mobile devices (such as pointing or scanning). That's why applications to improve the feedback and use of this interaction technique on tactile mobile screens have been developed, such as guide systems for tourists (Hardy and Rukzio, 2008), systems to enable feedback from touch screens (Kaaresoja et al., 2006) or systems to provide blind people with touch interactivity (Kane et al, 2008). There are also attempts to generate wearable hand gesture simplified interaction, based on colour recognition of finger tips to interact with elements projected on a surface, acting like a limited multi touch fingertip based interaction system (Mistry and Maes, 2009).

Technological advances and the advent of tablet PCs generated new collaborative interaction software that also supported external objects to interact with the touch surface besides the hands. The addition of multi-touch technologies to virtual environments using portable devices is an interesting area of research for HCI aiming at software development for entertainment and games. The use of vision based technologies plus the touching features of portable devices allow the development of software that utilises 3D information, allowing the creation of interfaces capable of supporting 3D interactions. The use of a camera to determine three-dimensional coordinates based on augmented reality and internal positioning devices (e.g. gyroscope, accelerometer, etc.) combined with the fingers' position over the touch surface, open up the possibility to use more complex interaction techniques in 3D virtual environments (Antle et al., 2011; Shirazi et al., 2009). This type of interfaces can be used in educational environments, training and testing areas and information management and interaction (Peltonen et al., 2008). The main challenges for these technologies are related with the definition of the interaction and the communication between devices, to allow a fluent collaborative work between users.

Despite the significant advantages multi-touch systems provide there are several issues that have to be addressed when software is designed for these devices (Davidson and Han, 2006):

- Graphic context is crucial in order to develop software that operates using multi-touch surfaces because the design of these applications must be created to resemble reality as much as possible to improve user experience. The lack of three-dimensional visualisation can be an issue and limits the types of applications that could be supported by these devices.

- The gestures interacting with the graphic elements must be natural, providing an abstraction of a natural process, such as touching to select a specific graphic object. Several gestures are already well defined for these devices, but more complex conceptual actions, such as rotations, need to be defined depending on the context of use.

- The interpretation models or system metaphors must be able to use all degrees of freedom provided by the system, those are not necessarily the ones in real world. In that way, users can perform tasks intuitively according to their working area. This aspect is especially important when 3D real world interactions are involved as part of the design.

- The possibility of using both hands allows the addition of new actions and commands to the system, increasing the possibility of generating "collisions" between commands (i.e. a two fingers rotation and a two fingers resize can be confused by the system and generate a result different from the one expected). However, with an appropriate definition of gestures and how they are performed, this problem can easily become an advantage by providing more complex functionality.

- The structure of the systems must be flexible and provide the possibility to extend and modify their functionality for different surfaces, improving the level of interaction. Furthermore, it could be desirable to give the users the opportunity to create their own interaction patterns offering advanced flexibility.

Another aspect to consider about the development of these systems is related to design techniques and how to control certain environmental factors that could interfere with performance, such as occlusions, wrong user identification, ergonometric aspects, positioning of the light source and video capture techniques to interpret gestures (Ryall et al., 2006).

Nowadays the components in portable multi-touch devices have solved the light source occlusion problems completely via the use of electronic based surfaces, but in cases of larger displays such as table tops, there are still technical issues. These issues are mainly related to light sources that can be solved with similar technologies as in the portable devices but at an increased cost.

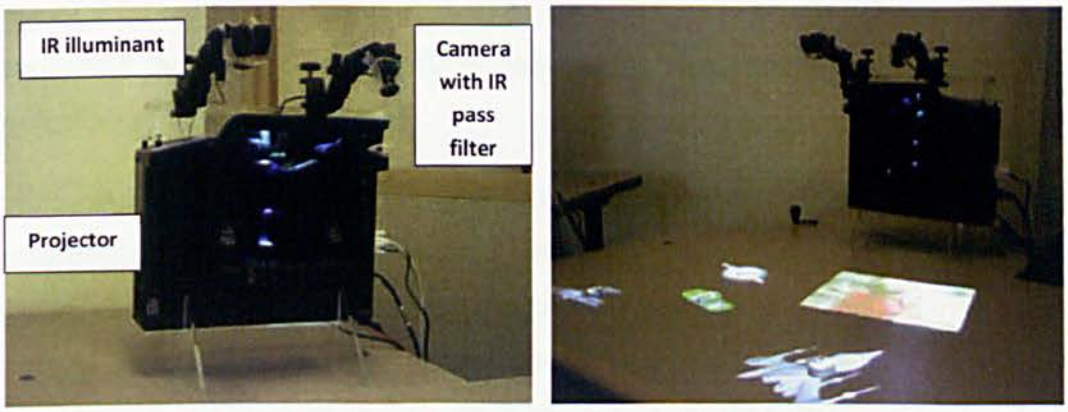


Figure 2.4: PlayAnywhere portable multi touch table-top prototype (Wilson, 2005).

One of the main physical limitations of the multi-touch table-tops devices is their lack of portability as they are generally large in comparison to their traditional counterparts. Advances in size reduction of the components of these devices (such as processing systems, cameras, digital projectors, etc.), enable the future creation of portable multi-touch table-tops, improving the chances of mass use (Wilson, 2005). The process allowing size reduction considers the use of smaller projectors and cameras, distributed in specific areas to cover the whole interaction surface, resulting on size reduction of the device, (see Figure 2.4).

Another aspect that can limit the usability of this technology is related to the 2D interaction method. The large as well as the portable displays, only provide two dimensions to interact with objects based on gestures directly over the surface of interaction. There are several studies in this area that attempt to enable these systems to operate with 3D interactions. This approach shows a significant increase in the number of interactions based on natural gestures hence giving the users more confidence and comfort when they use these devices, improving learning time and efficiency. One example of that is the Z-Touch (Takeoka et al., 2010), a system capable of detecting limited depth information enough to get finger positions, angles and distances from the touch surface (see Figure 2.5). This system uses an array of lasers, providing a new form of touch interaction. This approach shows a clear improvement in user learning speed and system efficiency and at the same time generates new challenges (Malik et al., 2005; Benko et al., 2006). These challenges relate to the interference with the light source and the potential occlusions due to the use of infra-red light detection devices in 3D space. Only simple 3D gestures were supported therefore further work is required in this area to fully explore the potentials of the third dimension.

Another example of surfaces that support touch interactions and 3D manipulation is the work on tangible interfaces, where front projection systems, lasers, objects and augmented reality can provide multi-touch interaction over a pre-defined surface. This tangible interface technology is a middle point between 3D hand gesture and multi-touch interaction approaches (Underkoffler and Ishii, 1998). A similar system is used in the work related to geographical interaction systems (Ratti et al., 2004).

Furthermore, it is worth mentioning the attempts to design 3D displays based on different forms, such as spherical (Benko et al., 2008) and cubic displays (de la Riviere et al., 2008). These devices are capable of generating a mapping of 3D data with 2D gestures overs the surfaces (i.e. the movements over a cubic-shaped surface are mapped on a 3D plane; however, the interaction itself is performed by direct contact with the squared surfaces).

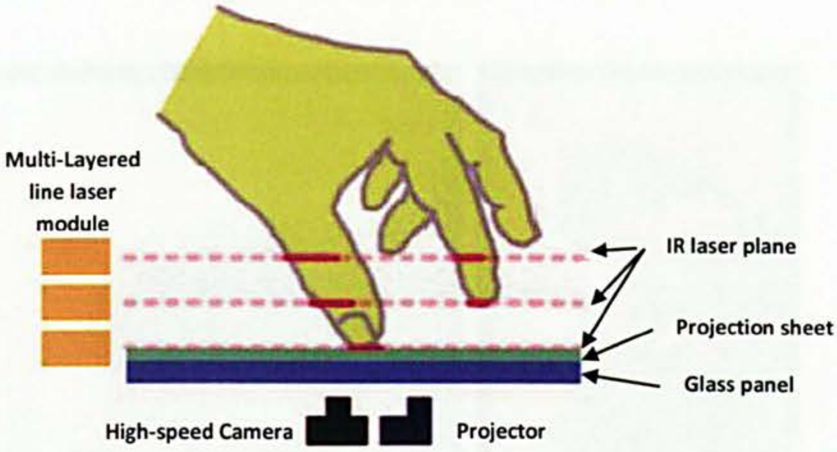


Figure 2.5: Z-touch 3D multi touch interaction diagram (Takeoka et al., 2010).

It has been observed that an important issue with these interfaces relates to the transition from the classic 2D WIMP interaction to touch interaction. In order to make this transition less problematic a connection with mobile interfaces was considered. An example of this is the research around interaction with mobile devices and multi-touch table tops. Here it was demonstrated that by using an integrated system it is able to increase functionality creating heterogeneous systems, capable of sharing information between users, (Döring, et al., 2010) providing a new way of multiple user collaborative interaction. Additionally integration with external devices is another area of development and research attempting to improve touch-based interfaces. The work on modifying traditional interaction devices to change and improve their performance and incorporating multi-touch features increases the possibilities of integrating multi-touch systems to traditional interactive systems, such as keyboards and mice (Villar et al., 2009) including devices that can generate 3D manipulation, such as the multi-touch mouse in Figure 2.6. This is considered an initial step in the early introduction to multi-touch technologies in workplaces. However, this new multi-touch hardware keeps the separation between the manipulation process and the element being manipulated from having the same problem as traditional input devices.

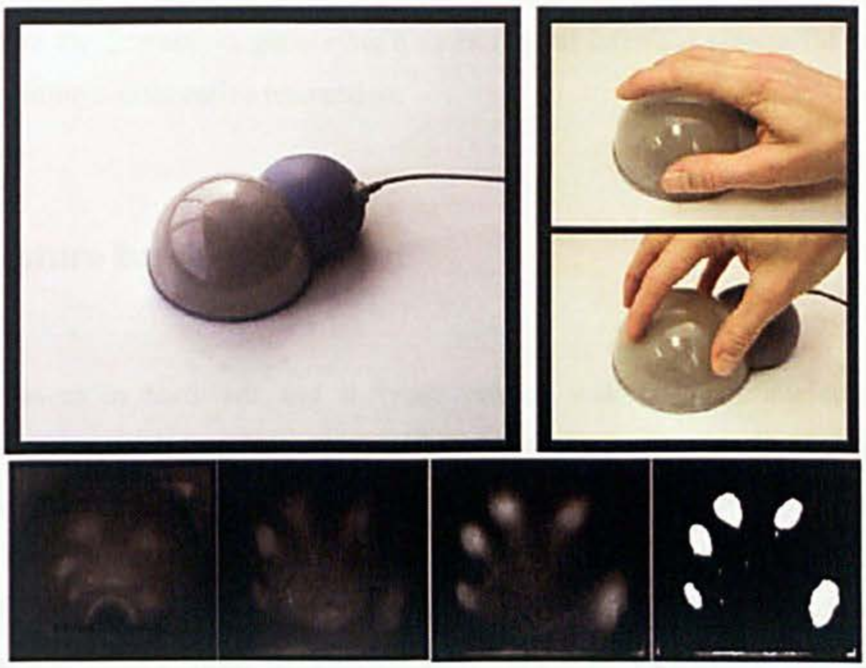


Figure 2.6: Mouse 2.0 multi finger interaction system (Villar et al., 2009).

A new research field related with these technologies is the development of software and table top systems able to manage not just multi-touch interfaces, but also other modes of interaction, such as natural language and integrating multimodal interaction in a collaborative environment. This integration increases interaction capabilities and the usability of these devices (Tse et al., 2006), but other challenges related to audio based devices, such as voice identification, natural language understanding, voice identification, noise, have arisen.

The impact of touch technologies is widespread and their use is becoming part of our everyday activities. Considering the future perspective of these technologies, the next step is concerned with integration with other media and devices that could be of the same type (touch) or totally different ones (traditional interfaces) plus increasing usability, functionality and efficiency.

Given the importance of the use of multi touch technologies, the extension of its use to portable collaborative systems is an important research topic that must be addressed and researched. The use of portable multi touch devices to create composite interactive system, when several devices can work as one, present an interesting research problem that must be addressed. The addition of 3D technologies to improve the interaction, generating a more natural interface system for users and also providing collaborative interaction.

2.2. Gesture based interaction

The advances in hardware and software created new ways of interaction with machines. Essentially, these advances are aiming to improve human-computer interfaces and their main objective is to make them more natural, based on body motion understanding, gestures and sensory integration (Blackwell, 2000; Francese et al., 2012). The area driving advances in interaction mechanisms is the video game industry due to the need to provide new levels of experience and advanced interaction between the users and the systems. Innovations in the game industry have been used in research areas (e.g. graphic processors, interaction devices, tracking methods, body motion capture, etc.) thus providing both new hardware and software based solutions (Barr et al., 2007). Hand-gesture interaction is one of the areas that can provide the mentioned objective of provide more natural interaction.

The first attempts to create hand-gesture interactive systems started with the use of haptic devices, such as electronic gloves, capable of retrieving information about the movement of every single finger of the user. The use of glove-based input devices was one of the first steps replacing typical interaction devices with more natural interfacing hardware. These devices operate in several ways mainly by tracking and identifying human hand movements, incorporating visual, audio and electronic techniques. The gloves provided that information using the appropriate hardware (e.g. LEDs for visual based gloves) (Sturman and Zeltzer, 1994). One of the examples of this technology is Nintendo's Power Glove (see Figure 2.7), which was a controlling device for video games and the cheapest alternative for most expensive and complicated glove devices. However, the glove-input technology does not satisfy the need of the user for more natural interaction. Even when several application fields were successfully explored and many interactive software systems were created, there were still concerns about "wearable" interactive devices. The main issues of these devices were related to use limitations such as degrees of freedom, restrictive or uncomfortable design and portability (Di Pietro et al., 2008). Also, the lack of positional information directly associated to the device was an important drawback of these gloves.

Figure 2.7: Nintendo Power Glove, created by Mattel.

Later new technologies were available that improved interaction by extending the degrees of freedom and included advanced position estimation hardware. An example of this are inertial sensors, such as gyroscopes, accelerometers and compasses that provided to typical interaction devices such as mice, the possibility to extend their degrees of freedom, (Olson, 1998). There have been several studies concerned with the improvement of the use of these devices in 3D interactive environments by including software solutions to improve their functionality and usability (Escudeiro et al., 2013).

There are other hardware devices that have contributed with the evolution of 3D interfaces. In order to provide natural interaction with computers wearable devices such as head-mounted displays (HMD) were introduced (Van Krevelen and Poelman, 2010), with novel studies on wearable virtual reality systems. In these cases the user is capable to virtually access a 3D environment and control computer generated objects (see Figure 2.8). This technology confronted several challenges, such as user position and depth awareness. Some approaches to solve these issues were based on complex mechanical systems, connected with ultrasound devices and separate Cathode Ray Tube (CRT) mini displays in each eye. Additionally, all of them were connected to larger devices to calculate the relative position and display the images, making these devices unwearable due to mobility constraints (Sutherland, 1968). Advances in hardware size reduction and inertial sensors, such as gyroscopes and accelerometers, have made this technology portable in recent years, allowing the user with a single device to interact both with 3D virtual and real environments using augmented reality enhancing the normal sight of the user, (Liu et al., 2010). Consequently new advances in wearable displays, such as Google Glass and Oculus Rift may increase the use of 3D interactive applications significantly.

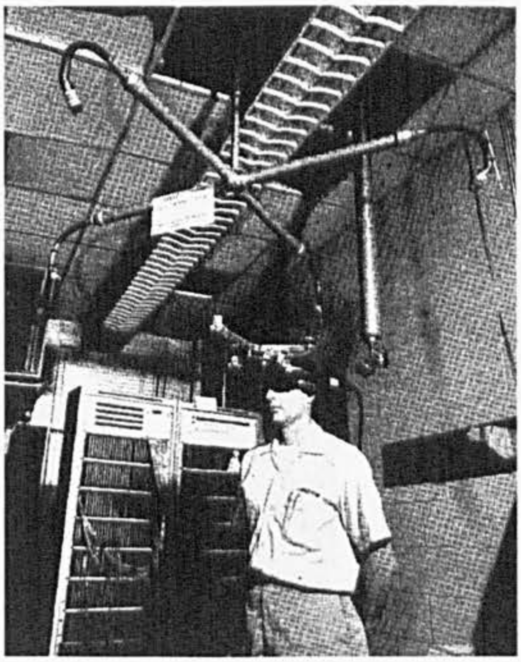


Figure 2.8: Mechanical head position sensor with head-mounted display (Sutherland, 1968).

Attempts to generate interaction in virtual environments were reported using stereo head-tracking systems that support single finger interactions determining the pointed and touched elements (Teather and Stuerzlinger, 2011). The drawback of these systems lies in the use of complex and uncomfortable pieces of hardware, limiting their usage on indoors applications.

Based on the previously mentioned advances, the video game industry has improved the user experience and interactivity by combining computer vision techniques. One example of these advances is the WiiMote, an interaction device created by Nintendo which allows the user to freely interact with video games (see Figure 2.9). The technology behind this device combines accelerometers and infrared cameras to calculate accurately the relative position of the device in relation to the screen. The communication with the console (Nintendo Wii) is based on a Bluetooth ad-hoc network. The low price and accurate position retrieval make this device an interesting and highly recommendable alternative to test and develop 3D user interfaces based on single point 3D gestures (Schlömer et al., 2008; Wingrave et al., 2010). However, as was mentioned, this device provides only one point of interaction without being applicable in more complex 3D interfaces.

Figure 2.9: Nintendo's Wii Mote (Schlömer et al., 2008).

During the past few years the use of infrared systems for 3D interaction proved to be highly effective to provide more natural interactivity between humans and computers. Improvements in computer vision methods for the detection of movement and depth contributed to the development of the next generation 3D gesture based interfaces using infra-red light projection. A device of this type that has been used in gesture based interfaces is Microsoft Kinect (see Figure 2.10). This device provides a low cost alternative to expensive laser based scanners and other similar technologies, allowing a wider range of studies in areas related to 3D real-time modelling and natural interaction. This sensor is capable of capturing depth and colour images simultaneously and integrating both sources of information to generate a fairly accurate representation of the captured scene. Also, it consists of several components such as microphones and a motorised tilt system, to change the device's orientation although its real contribution is the vision related features. This hardware consists of an infrared laser emitter, an infrared camera and an RGB camera, where the depth measurement is calculated by the triangulation of an infrared pattern, generating a full view of the environment in the working range of the device (Khoshelham et al., 2012).

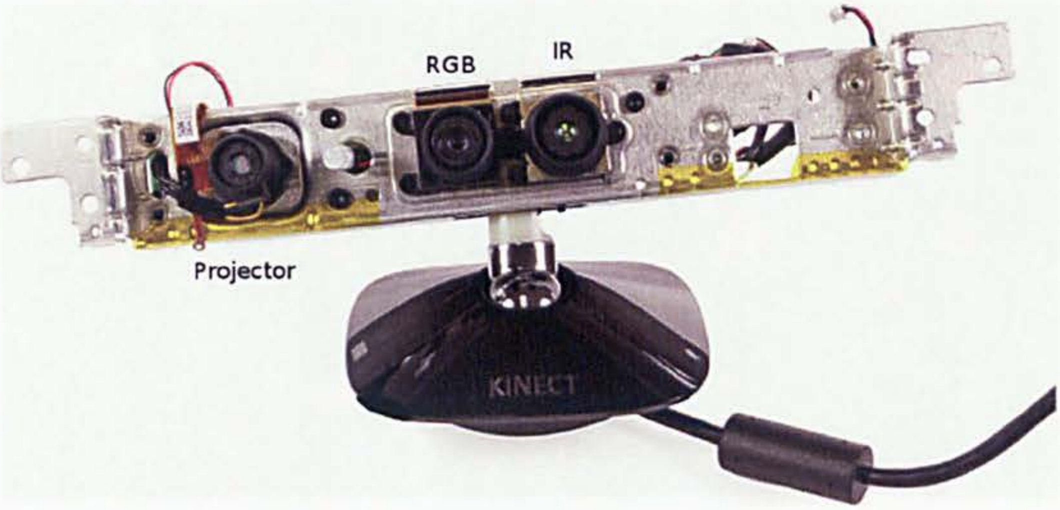


Figure 2.10: Micorsoft's Kinect vision system (Smisek et al., 2013).

Kinect however, tends to have noise related problems due to the infrared data capturing system generating some discrepancies about the specific depth distances and disparities in large range applications (more than 3 metres from the device). Its best performance and accuracy is achieved at shorter distances (i.e. between 80 centimetres and 3 metres from the device) (Khoshelham, 2011). This is the main reason why this device is a good alternative for hand gesture based interfaces. Studies performed in hand gesture recognition using Kinect to interact in 3D environments have shown promising results. An approach that could be used to recognise hands is based on Red, Green and Blue (RGB) skin colour detection including depth information (Tang, 2011) identifying the hand as a whole element and two states, open and closed providing the required elements to manipulate objects. Furthermore, fusing depth information and tracking algorithms, systems using Kinect or similar devices are able to detect finger tips and use them as independent input sources for gesture-based applications (Ren et al., 2011; Raheja et al., 2011). Researchers achieved full hand reconstruction in 3D (see Figure 2.11), including fingers, joints and their respective position and movements in real time (Oikonomidis et al., 2011). This approach however, requires powerful hardware systems and Graphical Processing Unit (GPU) acceleration methods increasing the overall cost of the system.

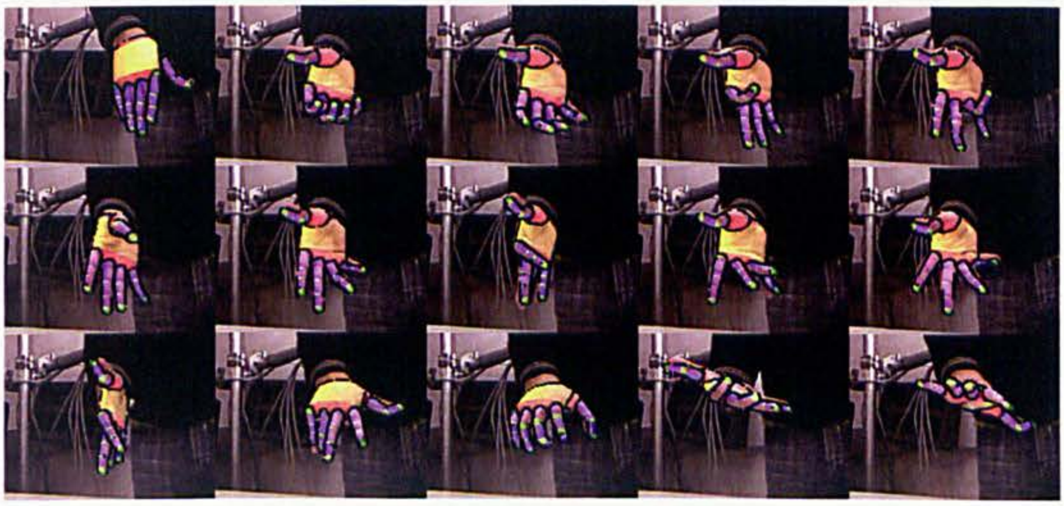


Figure 2.11: Hand articulation real time tracking and reconstruction using Kinect (Oikonomidis et al., 2011).

Nowadays, there are several other devices with similar features as Kinect. These devices have hardware for image and depth acquisition and operate in a similar manner combining RGB and infra-red technologies. Examples of these devices are the 3D sensors developed by companies such as PrimeSense, with shorter range of interaction and are half the size of Kinect. Another example is the LeapMotion controller, officially released in 2013 and contrary to its predecessors, works vertically and captures the lower part of the hands to generate the gesture. In 2014 the Kinect 2 was released and was much improved over the original, providing more accurate detection of user movements, including built-in hand recognition capabilities.

Despite advances in hardware and software to generate full gesture interaction systems, full 3D hand-gesture interfaces still require significant improvements both in design and development in order to improve a number of aspects. For example a salient issue is the transition from keyboard and number pad based interfaces that are implemented for 2D interaction in the real world to an equivalent 3D representation. Another pending challenge is the standardisation of 3D hand-gestures for interaction that are essential in establishing a common language of interaction (Pavlovic et al., 1997, Wu et al., 2005).

The use of gesture based interaction, as was shown, can improve significantly the user experience and the interaction capabilities in computer systems, allowing the generation of better interfaces and improving the overall human-computer interaction. The previously mentioned fact makes the research crucial. The use of fingers and hands combination to generate natural interfaces, using depth capturing devices such as Kinect can help to overcome to issues previously mentioned, especially reducing costs to create interactive hand gesture systems that may provide simplest tools to generate standard interfaces. Additionally, this method of interaction can be applied to visual programming system, since there are non-graphical components that make that software development more difficult and less intuitive (Clerici et al., 2009). These components have not been advanced enough to provide the necessary flexibility and clarity to understand many aspects of the development process that could be better grasped with a full 3D graphical user interface (Wachs et al., 2011). These problems become more obvious in pure graphic applications that require a better understanding of the environment where the tools will actually operate (Rotard et al., 2005). The research led solution to this problem prompted the creation of other kind of interfaces, such as 3D hand gesture based environments (Chaudhary et al., 2013; Murugappan et al., 2013).

2.3. 3D environments and data interaction

During the last few decades several studies related to 3D interaction in real time have been performed. These studies have contributed to numerous advances in hardware and software and have improved interaction with systems. These advances in software have helped to understand human gestures and movements. Tools to analyse and produce representations of real spaces able to be manipulated by users have been created (Stodle et al., 2009). 3D visualisation and representation is used in several areas nowadays especially in entertainment and design, where this technology has demonstrated the possibility of improving different aspects of human-computer interaction, opening new spaces to generate collaborative work.

The advances in 3D interaction would not be possible without the advances in Augmented Reality (AR). During the last decade augmented reality and its applications started to attract many researchers and was characterised by some fundamental properties (Azuma et al., 2001):

- Supplements the real world with virtual objects (computer-generated)
- Runs interactively and in real time
- Alignment of real and virtual objects with each other.

Augmented reality technology belongs to the "mixed reality" techniques and is regarded as the pre-processing step of augmented virtually systems. AR can be potentially applied to all senses not just in visual environments. There are several advances of tactile augmented reality, especially when it is applied for the treatment of phobias (Carlin et al., 1997). Also in case of "audible augmented reality", a more realistic experience can be generated by linking visual augmented reality and audio (Rozier, 2000).

The augmented reality technology started in 1960s but its real growth was during the 1990s, when several mixed reality systems were presented. There are several related problems to solve in order to achieve full functional interfaces including image and scene resolution, user's field of view, light contrast and lack of brightness in a real environment. Some of these problems could be overcome by the use for example, of optical see-through displays. There are other aspects related to the user viewing orientation and position. Even when the problem for known environments is almost solved, the case of unknown environments, such as open fields, is still a challenging area especially for outdoor applications where the lack of baseline information (such as specific elements or predefined markers to indicate relative position), increase the number of challenges to overcome. Finally, the calibration problems that were a main issue at the birth of this technology are reduced using several techniques, such as perspective projection models, position awareness hardware and auto compensation of calibration parameters. Still, few studies have addressed the advantage of the previously mentioned system to improve the quality of augmented reality. Advances on tracking, interaction and display technologies, however, keep improving the results and applications of augmented reality systems (Zhou et al., 2008). Since augmented reality can fuse 3D virtual objects in 3D environments in real time, several studies and approaches were introduced to make this technique more effective and efficient. The main application areas are in environments where additional information could be crucial in taking decisions, such as medical visualisation, maintenance and repair, military aircraft navigation and targeting (Azuma, 1997).

Another issue that should be considered is how the data will be visualised. For an environment that aims to be more natural for the user the most appropriate option is to present the virtual environment in 3D. The creation of this environment is important and the lack of tools required generating these types of interface increase the complexity. The work of Esnault (Esnault et al., 2010), addresses the problem of how future Web3D can be generated reusing design experiences and by separating the data and their representations. The approach used by Esnault is to divide the system in two substructures: the Genotype, a structuring metaphor construction which contains all the logical elements, such as data structure definitions, where the models of information exploration and the mechanisms to access the data sources are defined, and the Phenotype, which defines the visual aspect of the metaphor construction and the 3D visualisation of the interfaces. The system is constructed using style sheet techniques, web-based components and an intensive use of XML. The separation presented allows strong reusability of both components in future systems. The main drawback of this method is due to the lack of available tools needed to develop 3D environments. The interface has to be created using common 2D elements in typical development systems. In addition the system proposed for the 3D environment is just for web interaction and it is not developed for natural interaction and developing systems.

There are several advances in video game technology that have been widely used in the improvement of 3D environments. The development of new techniques on 3D rendering and multi view modelling in video games plays a critical role in the improvement of 3D technologies. For example games that are based on first person interaction or multiple camera view displacements, incorporating artificial intelligence engines in the construction and use of realistic environments provide advanced mechanisms for testing and evaluation in several HCI research areas (Andreoli and De Chiara, 2005). The influence of video game technologies lead the research on interaction environments, generating a cross evolution between different knowledge areas (Poole, 2004).

In addition, the use of modern video game devices that initially were developed for entertainment, changed approaches used in HCI mechanisms mainly due to their 3D acquisition capabilities, introducing several new applications. The work presented by Tang (Tang, 2011) relates to the Kinect utilization for hand gesture recognition and in Fratti's work (Fratti and Prattichizzo, 2011) it was proposed to use the same device in the field of wearable haptic technology to improve the response and detection of movements and gestures. This research has allowed the recognition of hands and fingers, the only issue is that the tasks presented are oriented to the manipulation of images in a 3D graphical environment without aiming for a higher level of interaction. These issues have been addressed by several researchers and new technologies based on different interaction devices have been developed (Van der Bergh and Van Gool, 2011; Noguera and Torres, 2013; Coelho and Verbeek, 2014).

Interaction mechanisms with 3D scenes have been always a challenging topic for image and video processing. Difficulties arising from these tasks are attributed to the several factors involved in a full 3D real time multi view-point processing, such as representation, capturing, rendering and coding. In the work of Tanimoto (Tanimoto et al., 2011) an approach aiming to deal with all these problems was presented, generating the first efficient 3D free viewpoint TV system. This method is able to show any arbitrary 3D view of a live scene without losing the interconnection between the viewpoints, generating a complete 3D reconstruction of the captured motion and providing the option to freely view a real scene in any given position and angle. The Free view-point TV (FTV) system consists of several components:

capturing, correction of views connection in a scene, rendering, user interface and coding. The depth estimation and interpolation between different views allows connecting a limited number of views to provide visibility to covered areas that permit infinite 3D views from a single scene. There are two approaches regarding the user interface for FTV. In the first case, the user viewpoint is based on head tracking to generate the corresponding view image, providing the capability of seeing any possible aspect of the scene generated and giving a full 3D experience of a real environment captured by the system based on the users' movements and their relative position to the environment. In the second approach, the user can freely change the views allowing the user to obtain any desired perspective in real time.

The possibility of representing a large amount of information in 3D was explored in detail in the work of Marcus (Marcus et al, 2003) where the representation of a large software system using 3D models allows a better understanding of higher abstraction level data. The most significant aspect of this system is related to the user interaction and the 3D visualisation of nested levels of code. Each element represents a code segment (e.g. containers) or information sources (e.g. cylinders) mixed in a map that can be manipulated and viewed in different positions. Also, it provides the option for 2D visualisation of elements presented in several categories such as functions, control modules, data type definition, variables, etc., which are represented in different colours. The major disadvantage of this particular design is that the interaction is still in 2D and the traditional devices that are used retain the disadvantages of 2D interaction in a 3D environment. The interaction and the use of more complex manipulation commands or combinations of them are still limited in this model of representation since it depends on traditional interaction methods and devices.

Even with these advances, the design of systems to program, develop and interact with 3D elements based on hand-gesture interactions, is still a challenging task. There are several examples of areas that can benefit from this technology, especially related to data and interactive manipulation on virtual and augmented reality environments. Medical imaging particularly in the area of manipulation and interaction with data such as proteins and MRI was benefited by the use 3D representations, reducing the processing and manipulation time (Böhm, 1998).

An area that can be significantly improved by the use of 3D interfaces is data manipulation. The multidimensional representation of information is not a new area of research (Stefanidis et al., 2011). There have been multiple efforts associated with managing databases in more than two dimensions to improve methods of information retrieval and data modelling (Agrawal et al., 1997). These representations of information are based on queries on multi-dimensional databases (related with several tables) such as On-Line Analytical Processing (OLAP, a technology for data warehousing that allows multidimensional representation of relational databases using cubes to organise data (Chaudhuri and Dayal, 1997; Zhao et al., 2011). An interesting aspect that is related to 3D databases is cube modelling and all the possible applications of that model. The possibility of representing multiple sources of information as a unique tri-dimensional entity, provides the ability to manage data that could be impossible in traditional interpretations, allowing relationship management and mapping of "hidden" information (Vassiliadis, 1998), especially useful to discover unseen relationships between information sources. Actually all of these models are managed under traditional interfaces, for example command line interaction is used to create and manipulate all of these models and simple graphical representations (e.g. disconnected tables). Instead, using a 3D graphic model to present these data, a cubic representation is more natural to interact with multi-dimensional data (Lancaster et al., 2010; Gomez et al., 2012). A 3D representation and a 3D interaction method for these cubes will provide a more intuitive system and a better understanding of how the information in "each side" is related with the whole dataset. The possibility of visually interacting with this cube (i.e. selecting rows, rotating sides and retrieving information) using just gesture based interactions will increase the productivity and efficiency in manipulating and modelling all these type of entities reducing the required learning and training time.

In terms of aiming to improve the development process of software, 3D gesture based interfaces can be used for robot action control. Development of systems and software for robots is an area related to multiple fields and technologies. The main aim of robot programming is to make them perform specific tasks in an efficient way mainly because their use could potentially not just be confined to experts. Visual controls to manipulate specific functions of robots are popular and largely used because they provide several advantages in the manipulation of specific components and functions, such as displacement and articulation movements (Corke, 1993). Even that several graphical interaction tools have been created, they still use traditional interaction techniques with all the drawbacks associated to the limited unnatural interaction methodologies. A gesture-based programming approach can improve this task, especially in humanoid robots, where the interaction needs mimic human reactions, actions, and interactions (Kanda et al., 2002). The methodology of Robot Programming by Demonstration (RbD) allows humanoid robots (such as HOAP-3) to learn movements and gestures using wearable sensors (Canlion and Billard, 2007), that could be removed by the use visual sensors as Kinect (Chen et al., 2012).

There are other areas that can be improved by the use of 3D environments for development of buildings. The modelling and development of an electric system for large buildings are areas studied by Kersting (Kersting, 2012) and could benefit from 3D developing interfaces, due to the increasing need to make them more efficient and reliable. 3D visual models of electric implementations of real buildings, based on 3D graphic components that resemble the real ones, allow the integration of critical structural information providing several points of view of the electric system and prevent possible risks in the real world implementation. Also, the integration with specific devices, such as temperature control systems, in a 3D environment could speed up the construction of new buildings (Oldewurtel et al., 2010). A gesture based system may improve the collaborative and interdisciplinary work highly necessary in the construction and planning of electrical systems for large edifications making the whole process more intuitive and reliable.

The exploration and tracking of elements in 3D industrial scenarios has also benefited from the use of modern techniques such as reconstruction and sampling, improving the general control systems of industrial equipment (Simoes et al., 2013).

With ever increasing advances in high demand graphic interfaces and multiple user applications there is a higher level of computer processing capability. The advent of multicore hardware architectures and parallel computing allows desktop and laptop computers to carry out multiple tasks concurrently. Even when hardware is crucial to achieve the previous mentioned goals, the possibility of multiple process execution at the same time would be impossible without the appropriate software implementations. Multithread programming has provided software systems the possibility to execute multiple tasks simultaneously, (Nickolls and Daily, 2010). However, writing multithread programs is still a difficult task since an intrinsic complexity is derived from generating multiple executions of tasks concurrently making it a far more challenging task than traditional sequential programming. Problems related to concurrency errors, including deadlocks (a situation in which two or more competing actions are each waiting for the other to finish, but neither ever does) and race conditions (situation where the output of an operation is dependent on the sequence or timing of other uncontrollable events) are caused mainly by the non-deterministic nature of the thread's order of execution. Testing the software has proved an inefficient way to deal with the "Heisenbugs" (software bugs caused by non-deterministic conditions), and that inefficiency has led to solving the origin of the problem: eliminating the non-deterministic characteristics from the multithread software, and forcing all the execution of the software to generate the same result given the same inputs. There have been many approaches lead to deterministic parallel programming, with successful results, providing the capability of isolating threads to generate the expected results in multithreaded programs (Liu et al., 2011). Nevertheless, there is a still a basic problem in the creation of multithreaded software: the visibility of the code created itself. In the text-based environments for programming, tools are used to assist in the process of creating and controlling the execution of the threads (such as controllers of the thread's execution order, monitoring and overlaying techniques). Also, it should be mentioned that even 2D graphics are not enough to provide the programmers an environment that avoids the problem of concurrent developing (Lee, 2006). The use of three-dimensional representations using graphic features appears as an alternative solution to the visibility problems of parallel coding (Wong et al., 2012).

The creation of flexible 3D frameworks to generate interactive 3D environments, allowing reusability of component and providing fairly easy to understand data interactive interfaces presents a challenge that must be addressed in future research.

Database graphic interfaces correspond to graphic user interfaces that allow direct interaction with information elements stored in a database (Schaefer et al., 2009). The database graphic interaction can be achieved by the use of different methods, including gesture interaction, which corresponds to interaction performed by the user based in body motion (typically using the hands and fingers), allowing graphic elements manipulation (Epps et al., 2006).

Devices that provide non-traditional interaction methods with data, including devices such as tablets, smart phones, gesture-based systems such as Kinect, and eye-tracking-based systems (such as Google Glass) are becoming more popular rendering their predecessors outdated. Consequently the development of next-generation user interfaces for data interaction has become essential. Given the amount of data that can be processed using non-traditional interaction devices, the Query-Result paradigm (where the user types a query to the database and waits a specific amount of time to get an answer, depending on the complexity of the query) becomes inefficient. Interactive tools are limited by using back-and-forth paradigms (where the user has to wait for the result after the interaction, not getting immediate feedback from the interface; e.g. when searching in Google) (Nandi et al., 2013). The development of new ways to interact with data and more specially, to retrieve information from data interfaces such as forms, reporting tools and query workflows generates new challenges: query latency, databases workload under a graphic context, triggers setup, interface intuitivity and feedback generation. Querying interfaces that are addressing these challenges are slowly appearing (Jiang et al., 2013).

QWiK (Nandi, 2013) presents an interface for databases using different types of natural interaction including touch and gesture based interfaces, focused on multi-touch interaction. The graphical process allows dragging of two data tiles in the same direction; the interface detects the movement and the corresponding table elements, offering possible queries. The available set of possible queries changes according to the user's selection of attributes and finally performs the join operation when the user drags the two attributes together, displaying the information of the operation. At an internal level the queries are treated as a probability distribution over all the possible space of the database until the gesture is finally performed, working inside of an event session that finishes when the final query is performed (the dragging process between the fields of the two tables), narrowing the space of possible queries to just one. The internal architecture of the system is of block-based modular architecture where the user interface and the gesture mapping components are independent of the QWiK core. This provides the possibility of being changed without altering the internal system's composition. In general terms, the system's core can be divided in two basic parts: the Intent Interpretation module in charge of determining the most likely query given the gestures performed, and the Feedback Generation module in charge of generating insights and the final results of queries. Even when this system provides an interesting solution to the problem of generating graphical queries over a database, the use of a simple relational model drastically reduces the chance for more complex interactions and three-dimensional interfaces. Figure 2.12 shows the QWIK internal architecture

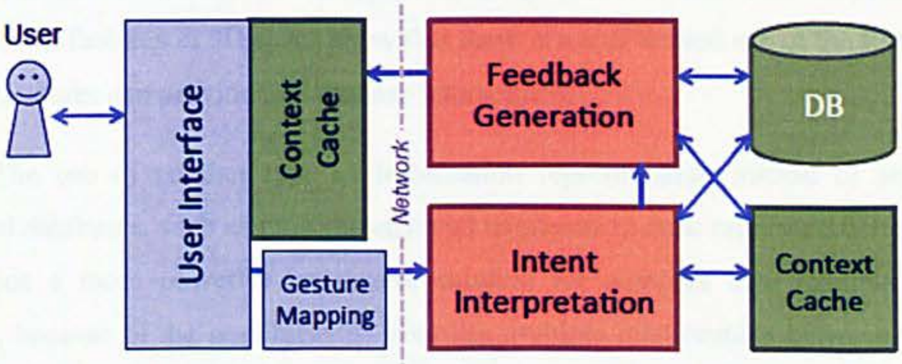


Figure 2.12: QWIK's internal architecture (Nandi, 2013).

Other areas of research on data manipulation focus on different types of information, (i.e. images, shapes, 3D geometric figures, 3D representation of real objects, etc.), where motion sensing devices, such as Kinect, are utilised providing a natural interface for direct graphic data interaction. One example of that is "The Kevin Bacon Game" using a navigation system on Internet Movie Database (IMDB), a Kinect interface and a logic data stream engine (Bier et al., 2013). This system was constructed using the InfiniteGraph database (developed by Objectivity Inc.), a graph database system (i.e. databases constructed internally as a graph, where the data components are nodes and the connection between them are edges; without using the traditional relational model) that works under Java objects. The graphical construction of the database is based on a structure of nodes and edges, where each node is an object of the given class for the database and the edges provide the logical connection between the nodes, given a specific attribute. The interaction of the system was constructed using the OpenNI framework to detect the user's skeleton and provide functionality using the right arm as interaction pointer. The interaction mechanisms provided in that approach is a set of basic functions (select edge/node, obtain information of a selected edge/node, undo/redo actions, search the shortest path between two nodes and solve the shortest path between two nodes given a specific characteristic). The evaluation interface was based on the "Kevin Bacon Game", where the user has to graphically search the shortest path between Kevin Bacon and another actor (all the information is provided by a search engine connected to IMDB). Even when this approach provides a more interactive way to request information from a database, the limited graphical context and the poor use of the Kinect features in 3D space show that there is a still limited use of the features that 3D gestures can provide for database interactions.

The use of another type of information representation instead of simple relational databases, such as multidimensional databases in cube representation seem to provide a more powerful interactive solution for complex data manipulation systems, because of the possibility of visualise multiple relationships between data, allowing to organize and process information in a more efficient way. The mentioned advantage can be improved with graphic interfaces, especially if it is combined with a 3D modelling.

2.4. Development challenges in 3D

3D interaction mechanisms and systems appear to be the next step in human-computer interaction evolution, given the efforts to create more natural and real world-like interfaces, as it was discussed previously. Recent advances in hardware and cost reduction of fully 3D interactive devices has augmented research and development in this area. However, especially in programming and data interaction environments, there are important issues to be tackled at the design stage of new applications, mainly related to 3D interfaces.

Developments in interactive environments show the need of fully 3D interactive systems. During the implementation of graphical software, developers use visualisation tool such as Unified Modelling Language (UML) diagrams representing the ideas based on graphics accompanied with text. The graphical interfaces lack good representation of the real environment where the system will operate (Edmondson and Beale, 2008) making the understanding of the actual problem difficult during the first stages of development. If the development process was undertaken using the same metaphors of the final product, the understanding of the implementation tasks would be improved. In order to achieve that, more natural interfaces such as Tangible User Interfaces (TUIs) and Pose Estimation Systems (PESs), could be utilised (Kim and Maher, 2008; Erol et al., 2007).

Other issues relate to the traditional implementation of 3D interfaces. As mentioned before, the development of 3D interfaces is based mainly on 2D components (images and text based interfaces) (Yasuda, 1999). Creating a 3D interface is a complex task which requires significant amount of time, principally related to connecting different 2D views of the environment and then making them usable in a 3D context (Regenbrecht et al., 2001; Gu et al., 2010; Ullmer and Ishii, 1997). The elevated complexity associated with the manipulation of these views and their connection with the core system, which in many occasions is created without taking into consideration the application environment, makes the resultant interface highly prone to errors. This increases weak points in the tools that will therefore result in system errors or even system failures. Therefore, considering a full 3D programming environment could help to reduce these issues.

The reality abstraction representation aimed by the system is another issue that must be considered. Software programming in general, and especially programming based on paradigms, such as object orientation is based on an abstraction of reality. When a programmer deals with systems that will operate in three dimensions it is often necessary and desirable to understand the associated concepts. Frequently these concepts represent direct abstractions of reality; like address, identification data, description of characteristics, business management elements, etc. These concepts are not easy to understand or program when the only semantic tool that can be used is plain text and few shapes representing basic concepts (Kobryn, 2000). Moreover, when software development requires the understanding of abstract concepts such as data repositories or action handlers (such as opening or closing files and data streams, and reading info), the lack of good representation of them makes the developer's tasks almost impossible (Chen et al., 2000). Also, when those concepts are not clear the forms of interaction, tasks and even the aim of the environment can become problematic. Therefore, the user will have much more trouble using the application thus requiring learning of new metaphors not related directly to the working environment. These issues may lead to the failure of the product (Peters, 2004) therefore a well-defined 3D object representation applied in the right way could reduce these problems, providing useful visual information to the user and improving the general understanding of the problem.

Another related issue is the limited visibility of graphic elements and their interaction that are available under a 2D representation, particularly during the initial stages of software modelling. During these stages of development, the only method for programmers to view the result of their work is via testing specific sections of code using simplified prototypes of the whole system for specific graphic elements or generating some interfaces using a graphical development environment (Reiss, 2007). They do not have the possibility of a complete view of their work in real time. Not even a section of the code can be viewed whenever the developer wants (Itoh and Tanaka, 2006). Even though there has been significant progress in the development tool the lack of viewing representations in an interface (e.g. combining design, parameters and interface) limits the possibility to fully understand a problem and consequently, possible solutions during the implementation (Shim et al., 2005). A 3D software development tool would increase representation details and viewpoint allowing the developer to deal with these issues more effectively. Furthermore, 2D representations also limit the interaction with programming elements. Software developers work in an unnatural way because their interaction with the tools is limited to few movements using single input devices (e.g. mouse) instead of using hands, which can provide more interaction points and faster and easier manipulation, and can resemble real world interactions. In most human activities, it's possible to use more than one point of contact to perform common activities (e.g. grab or move objects) in order to accomplish more difficult tasks (Poupyrev et al., 2007). With an efficient 3D development system, the usability and learning time will be improved both for developers and users (Myers et al., 2000).

It seems that the most appropriate way to create software that has real world resemblance is by using a similar representation, (i.e. creating software using a three-dimensional interface). This suggests the development of 3D applications will be made simpler with a 3D environment. Furthermore due to the arrival of new hardware (e.g. Kinect) interaction mechanisms and the tools to create software have been improved thus supporting the previous analysis (Barr, et al., 2007; Marsh, 2011). However there are still outstanding aspects to improve in order to achieve software representations that can be utilised to create new systems. These aspects are frequently related with interface related definitions, such as how the user will interact with the system. Many attempts have been made to define intuitive interaction mechanisms, (Telea et al., 2010). Many of these new ideas operate just in 2D. Therefore the problem with 3D interaction and manipulation of objects remains associated with use of depth information and issues related with the integration of that information with typical 3D interfaces (Teyseyre et al., 2009). Interaction mechanisms are not intuitive and adaptable enough to accomplish the needs of the users and the developers. Humans would be more comfortable with full 3D applications but to develop those applications is very difficult with the current text-based tools, therefore there is a need for a completely new programming paradigm using a 3D Integrated Development Environments (IDEs).

Many new interfaces have appeared over the last few years such as touch interfaces or “tangible bits” (Ishii and Ullmer, 1997) for different kind of applications, supporting many ways of interaction in 3D environments (Ray et al., 2007; Julien et al., 2010; Neumann et al., 2009), but most of them target end users rather than developers. The research community did not contribute enough to the design of frameworks or tools for software development. This is an open field with many issues to overcome, such as interaction, graphic components’ definition and integration mechanisms for these components (Irawati et al., 2005).

Other issues that increase transition difficulty from typical 2D to full 3D interfaces are related to the methods used to separate data from representations in order to achieve flexibility and reusability. Until now all of the graphic parts of software have been well isolated from the data, and only few specific parts of the code were able to make a link between them (von Pilgrim and Duske, 2008; Kapec, 2010) Therefore it's necessary to define which and how data elements (e.g. variables, arrays or more complex types of data structures) are connected with graphical elements (Chittaro and Ranon, 2007). These definitions are yet to be well established and that is why it's imperative to create a way to generate software in a 3D framework with a natural interface that enables the visualisation and the establishment of a clear connection between data and representations (Telea and Voinea, 2011; Gill and Tomar, 2010).

The selection of appropriate metaphors to represent interfaces and programming elements is another challenge in the creation of full 3D software developing environments. Because of that it is necessary to develop a representation of basic elements to create software in order to give the developers tools (e.g. sandbox), that should be intuitive (associating the basic elements for software programming with comprehensible concepts) and able to be used in 3D environments (Hurtienne et al., 2010; Qian et al., 2011).

In many cases it is necessary to consider the similarity between the interfaces and actual real world solutions. The problems in creating a 3D interface appear mainly due to the specific functionality definition which means that the functions created for a 3D environment are too specific for one application and are not flexible enough to be used in other projects. Also the details in an environment that generate the connection between the metaphor and the data to be manipulated (Wu and Balakrishnan, 2003) can be challenging. The problem of working within a 3D framework lies in designing software components (such as control, presentation and hardware connection modules) simple enough to allow the creation of more complex ones (Fishkin, 2004). The developers need to be able to create the graphic objects that are required using basic and flexible elements that they are familiar with such as small sections of code for specific tasks (Conway et al., 2000). It is hard to design components flexible enough to be used in a 3D framework in order to allow the creation of any kind of interface. A 3D interface simplifies for a programmer the development process any kind of application (including traditional 2D applications), because as it was discussed previously, it provides a more natural way to interact with graphic elements related with interfaces and development elements (Schmid et al., 2010). Users would be more comfortable using 3D computing but traditional Integrated Development Environments (IDEs) make it hard to develop such applications. So, the alternative to overcome this problem is in the creation of 3D IDEs for the development of 3D applications. This possibility requires further exploration as is suggested in the works of Jia and Osawa (Jia et al., 2009; Osawa, 2006).

The advances in computer interfaces and data management provide new ways to interact and manipulate information. These advances have also influenced research to improve the way software itself is developed, generating programming languages that use graphical elements instead of text-based environments. The main goal of Visual Programming Languages (VPLs) is to improve the ability of the developer to express the program using higher level representation and logic graphical abstraction that allow a better understanding of, and interaction with the developed software (Burnett et al., 1995).

The advances on graphic interfaces have also influenced the way VPLs are constructed, aiming to improve the visual tools to interact with higher abstraction concepts. An example of these advances is the creation of VPLs to manipulate mark-up languages, such as XML-Oriented Composition Definition Language (XCDL) (Tekli et al., 2013). The aim of XCDL is to provide a language suitable for both expert and non-expert users, offering a wide range of possible operations on XML data over a framework based on Coloured Petri nets, providing the capability to create visual representations for data interaction systems (i.e. web data). The use of language is based on drag and drop operations by grouping them according to the shape of decision trees, with multiple inputs and outputs. The definition of the sequence of operations is based on arbitrarily chosen colours that connect inputs, operators (provided as graphical elements by the language) and outputs that have the same colour of the inputs and operators previously mentioned. The definition of the XML data types is performed by using dialogue forms that allow the user to add several information fields (such as identification, name, type and description for input or output). This language, even when it provides several advantages over typical XML programming interfaces, still has disadvantages especially in the manipulation of elements and visualisation based on basic 2D representations and interfaces heavily loaded with text. Even when drag and drop interaction can provide extensibility for natural interfaces (such as multi-touch and gesture based ones), there is still an extensive use of text based interaction since interaction is still preformed using mouse and keyboard.

The main advantage of visual programming lies in its capability of supporting visual metaphors of the intended application domain. The majority of modern VPL are based on two-dimensional manipulation and graphic components, but the use of 3D graphic elements could improve overall representation and overcome overlapping problems, such as intersections between edges in sequence diagrams or overlapping of components for parallel applications. However the process of generating specific languages for specific environments is not an easy task. The use of a generator framework can simplify this task by providing tools to generate 3D VPL for a specific context. This is the case of the Development Environment for Visual 3D Languages-DEViL3D (Wolter, 2012). DEViL3D is able to generate three-dimensional editors, based on a 3D canvas that can be used under typical 2D interaction methods or 3D input systems (including Microsoft Kinect). The interaction is based on a structure editor that allows the creation of language constructs (such as methods, action handlers, data types and any other tools necessary to define and create the underlying structure of a visual language) via direct insertion and manipulation over these constructs. These constructs can be modified and characterised after being inserted by using widgets (graphic primitive elements that can represent language constructs, i.e. spheres, pipe-shaped connectors and shapes that can be defined and edited by the user) to define constraints and functionalities provided by the elements. The syntax of visual language is defined by an abstract structure that describes the connection between the language constructs (this structure can be viewed as a tree of labels that represent the constructs). The concrete representation of the language consists of a set of graphical primitives previously created by the developer. Finally, the structure editor permits the keeping of programs syntactically correct. This framework generator presents an interesting approach for software developing using 3D VPLs, but is still in an initial stage. Providing the construction of molecular model languages (3D VPLs based on simple spheres and connectors, which can be used to represent class specific applications), without allowing the generation of new 3D VPLs for specific contexts and architectures, such as parallel programming.

The development of 3D interactive systems generates several challenges as it was exposed before. The development of systems that can provide more natural interaction, flexible metaphors to generate software and an appropriate use of 3D representation for data interaction are open research areas that must be addressed.

2.5. User Interface Evaluation

The evaluation of interaction interfaces is an important issue and several methodologies have been proposed and used. Some of the most widespread and accepted classic methods to evaluate software interfaces are heuristic evaluation, cognitive walkthroughs, guidelines and usability tests (Jeffries et al, 1991).

Heuristic evaluation is performed by experts in the specific context of use of the interface. These experts evaluate the interface based on usability principles. The number of specialists is limited (to no more than 10) and the evaluation time period varies from hours to weeks depending on the complexity of the interface. Also, this technique is known as the most informal method.

Cognitive walkthroughs (Grigoreanu and Mohanna, 2013; Mahatody et al., 2010) are based on the performance of a specific task, where the user is supposed to confront a problem solving process step by step. This is a task-based procedure without specific guidelines to evaluate the tasks and requires long term planning before the actual execution. Also, to improve the results of this evaluation procedure, it is necessary to perform pre-tests before starting the real experiments. The number of users varies and generally the users require a period of training that depends on the interface. Also, it is necessary for the constant presence of an evaluator to control and record the results of the actions performed by the user in the interface.

The guidelines group uses a specific set of questions (generally developed by a specific company, enterprise or institution related with the application area of the software) to evaluate the accomplishment of usability requirements. This method is meant to be used by software developers and assessors, therefore it is suitable when the development group is numerous and the assessors (which are generally other developers) are limited, because it allows to the assessor to check specific issues related to the guidelines, avoiding elements not linked to the evaluation, (Brown, 1989). An example of such a method is the one introduced by the company Hewlett Packard with 62 specific guidelines for their software development. This method requires complete understanding of the guidelines, the specific design of the interface and it is not suitable for untrained users.

Usability testing (Barnum, 2010; Bastien, 2010) requires a group of users to actually use the software's interface understand how it operates and perform one or several tasks. After this process, the user is asked to answer a questionnaire about the task performed and, in some occasions, they have to contrast their experience using the tested interface against a previous version or a similar interaction method. This approach is suitable for users when the participation of external reviewers and the training periods (where the users get familiar with the interface) are relatively short. In general, the questionnaire is based on different human factors and in the interaction process, evaluating qualitative and quantitative parameters. Qualitative aspects to be evaluated are related directly to the point of view of the user, such as how well the performance of the tasks is, how intuitive, how easy to manipulate and how comfortable the interface is. The quantitative parameters are related to the execution of the given task using the interface, considering factors such as time to execute the task, actions performed to achieve a given task, quantity of errors committed by the user and errors related to the interface itself. This method is highly suitable to evaluate new interfaces and new interaction methodologies.

There are other methods, such as feature inspections, pluralistic walkthroughs, consistency inspections and standard inspections (Roy and Pattnaik 2014; Petrie and Power, 2012; Nielsen, 1994), but in general, their features are not really different to those previously mentioned.

Technological advances in cameras and human action monitoring introduced an evaluation approach based on eye movements (Goldberg et al., 1999). This technique is meant to complement some of the previously mentioned usability inspection methods, such as walkthroughs, using the eyes' movement as additional information to determine the search behaviour followed by the users to solve selected tasks. The scan paths followed provide significant information about the efficiency of the user to solve the tasks and help to improve the layout of the interface. The idea behind this technique is based on the spatial focus of attention of the user, which can indicate how well constructed is the interface, in terms of element distribution. This methodology provides quantitative measurement of the form of gaze concentration showing how well placed are the interface elements, derived directly from the length of the visual paths followed by the user. More advanced techniques in this area can include head movements too. This technique is used to evaluate the layout of the interface and the relative positions of the elements on the screen, but not user related qualitative factors, such as the difficulty to understand a given task, how intuitive is the interaction with the interface or how the user feels with the interface. Also, this technique is not recommended to evaluate totally new interfaces, where the interaction mechanism is the main aspect to be considered.

The evaluation of 3D interfaces in multi touch mobile devices presents new challenges. Evaluation of 3D interfaces is performed in a similar way to normal interfaces, with a training period, a task performance to obtain quantitative evaluation results and finally, a questionnaire to collect users' feedback about subjective issues. The procedure is similar to normal usability testing and the added component is that the same task is tested under different interfaces, such as traditional console interfaces or a graphical button based interface (Fiorella et al., 2010).

Under the previous analysis, the technique more suitable to evaluate new interfaces under factors related to the interaction itself is the usability testing, because it is capable of providing qualitative and quantitative feedback from the users, even with a limited amount of test subjects. This provides reliable feedback and evaluation elements to improve the interface's definition and solve design problems, avoiding complications due to long times of training or hardware calibration during tests (Bastien, 2010).

2.6. Conclusions

In this chapter, a review of human-computer interfaces for next generation devices, including multi-touch and hand-gesture based systems, was presented. Also, a review of the evolution of these interfaces and devices was presented, including an analysis of different techniques and tools used to achieve the available styles of interaction. Also, the challenges related to new interaction environments for software development were discussed, focusing on the use of 3D interactive systems to improve the users' experience, reducing training times, and improving the users' general performance and satisfaction. The use of 3D interfaces with 3D interaction mechanisms, based on multi-touch systems or two-handed gesture interaction, presents an interesting alternative to traditional methodologies. The use of iconic graphic metaphors allows a better understanding of the general problem in different interaction environments, providing several advantages over the traditional text-based representation of information both for users and developers. Human computer interaction and the continuous evolution of interaction environments, where the representation of information tends to resemble the real environments, providing natural interaction could be considered the main future areas of research and development. Studies related to multi-touch interfaces in 3D environments to allow collaborative work and hand gesture based interface in 3D will help to understand better how these techniques can improve the user's experience, performance of different tasks and learning time in new work environments. Also, studies about how 3D IDEs can improve the work of developers are necessary, especially in the context of multi-threading programming. However, all these new representations and interaction techniques must be evaluated under realistic conditions with feedback from the users and quantitative evaluation to assure that the improvements over traditional interfaces and interaction techniques are really achieved.

In the next chapter, the development for multi touch 3D interfaces will be discussed, including architectural design and basic elements, and a novel framework for multiple device interaction will be presented and analysed focusing also on collaborative tasks.

Chapter 3

3D interaction for multiple mobile devices and tablet Pcs

3.1. Context and Overview

In the previous chapter, a review of human-computer interfaces was presented; including a review of the evolution of these interfaces and next generation devices with an analysis of different techniques and tools used to achieve 3D interaction. The challenges related to new interaction environments for software development were discussed, focusing on the use of 3D interactive systems to improve the users' experience, reducing training times, and improving user general performance and satisfaction.

The evolution of touch devices has provided a new opportunity to develop interactive software based on the use of the display area as an interaction surface, making easier the creation of more user friendly systems. Nowadays, touch devices are accessible to everyone, becoming part of several areas of social interaction. Devices such as mobile phones, video game consoles, laptops, etc. (Malik et al., 2005) are increasingly common. Moreover, the use of touchable interfaces has been demonstrated to increase and improve the understanding of software metaphors, increasing productivity and reducing learning time, triggering cognitive and affective factors that are not able to be accessed under the use of traditional interfaces (Jia et al., 2013; Steffin, 1999).

Another aspect of multi-touch systems that has been established especially for table top devices is their advantages for collaborative interaction applications, (Higgins et al, 2012). Interfaces that support control and interaction with information simultaneously by multiple people receiving feedback for their actions provide new opportunities for designing novel interaction mechanisms.

The use of multi-touch interfaces and other interaction technologies can improve collaborative tasks based on these devices. 3D representations on multi-touch interfaces have shown promising advances especially in displaying and manipulating 3D data (Martinet et al., 2010).

Another technology that can improve the previously mentioned features on multi-touch interfaces is augmented reality. In these augmented reality systems, the use of 3D visualisation of real environments may enhance user experience on collaborative interfaces mainly in applications related to mobile and portable devices (Wagner et al., 2012).

Mechanisms for 3D visualisation and augmented reality can be successfully combined with other technologies to generate new interactive interfaces for entertainment, design, and other application areas. This is the case for advanced 3D real-time visualisation techniques, such as the in case of FTV (Tanimoto et al., 2011) which combined with the use of inertial position information from the multi-touch device can generate new ways to develop interactive content and entertainment software.

In this chapter, a novel approach to interact with 3D environments and elements is presented based on the use of 3D techniques, augmented reality and portable computing devices, such as mobile phones and tablet PCs. The proposed methodology allows the user to interact and visualise elements in a 3D environment based on the concept of free viewpoint television. In the following sections we present a brief discussion on previous related work, our proposed methodology, an analysis of the implementation. Finally we will present the obtained results and the evaluation process discussing related advantages and disadvantages along with possible applications in different environments.

3.2. Previous work

Multi touch devices provide the users with the ability to control and interact naturally with elements using interactive surfaces (Kin et al., 2009). These two key characteristics provide the possibility to create advanced graphic interfaces, improving interaction between people mainly in the case of collaborative environments where multiple users can see and interact at the same time with the information presented on the multi-touch device.

Several studies have been performed to evaluate the suitability of multi-touch devices and interfaces on collaborative interaction. The use of multi-touch devices in educational environments has been widely studied and their capabilities of support collaborative learning in classrooms have proven that multi-touch table technologies have a clear advantage over traditional computer interfaces (Harris et al., 2009; Higgins et al., 2011; Dillenbourg and Evans, 2011). Moreover, there are other areas where these collaborative features can be used to improve interaction such as urban planning (Wagner et al., 2009; Chow et al., 2011) or building energy management systems (e.g. systems to provide design and evaluation support for implementing and managing configurations in buildings) (Neef and Ferranti, 2011) where multi-touch interaction provides tools to resolve conflicts and problems (such as design features, implementation constraints, etc.) in a collaborative way. However, there are issues that must be addressed in these interactive multi-touch interfaces, especially ones for interaction using multi-touch tables, where multiple users share the same interaction surface. One of these problems relates to occlusion of the actions among users over a table-top (Hornecker et al., 2008). This can be harmful in collaborative work generating conflicts between users. Techniques based on vision based hand tracking in multiple user multi-touch environments aim to improve user interaction experience and reduce occlusion by identifying each user and tracking their interactions with the multi-touch interfaces (Dohse et al., 2008). The main issue with techniques for hand tracking is that they are based on skin colour detection and therefore the accuracy of following the interaction of multiple users is limited.

Advances in portable touch devices have provided novel ways to deal with user identification problem in collaborative activities, allowing users' autonomy of interaction, peer action's feedback and activity synchronisation. However, collaboration scenarios must be clearly defined to design and develop correctly such interactive multi-touch collaborative systems (Hersovic et al., 2011). Applications related to video manipulation using mobile devices supporting touch interfaces can be used to create interactive collaborative systems, where multiple users can have real time feedback from other users over a personal mobile device (Boring et al., 2011; Ponto et al., 2011; Zhang et al., 2013). Nevertheless these approaches are based on 2D interaction and are disconnected from real working environments, where 3D representations and augmented reality systems could improve user experience and make use of position awareness features present on new portable multi-touch devices.

The use of augmented reality (AR) for collaborative interaction has been studied largely during the last decade. Since augmented reality provides the capability of enhancing real word visualisation, adding computer-generated information and graphic elements (such as 3D models blended in the real world), provides multiple advantages that can be used in portable multi-touch interface thus improving virtual collaborative workspaces (Carmigniani and Furht, 2011). Real time detection and tracking for mobile applications can be used to generate augmented reality environments, using computer vision based techniques (Wagner et al., 2010) but these techniques are not enough to provide an accurate interaction over a multi-touch user interface. The use of position aware devices on portable computers and mobiles, such as GPS has proven it's usefulness in outdoor applications for panoramic viewing (Arth et al., 2011). GPS devices have been used in experimental designs, among other inertial devices such as gyroscopes and accelerometers long before the advent of portable tablet PCs and touch enhanced mobile phones (You et al., 2001; Lang et al., 2002).

Nowadays novel collaborative interfaces for handheld devices based on AR and inertial sensors have been developed and introduced (Olson et al., 2012) although the combination of this technology with 3D interactive environments has not yet been fully explored. Despite this, research in this area shows promising results (Weng et al., 2013). 3D visualisation plus augmented reality based on spatial awareness on mobile devices have impulsed the development of new interactive interfaces, enabling several users to interact over the same physical space with virtually generated 3D objects, especially useful for gaming, design, entertainment, etc. (Sohdi et al., 2013).

The work presented in this chapter aims to demonstrate the feasibility and convenience of 3D interfaces combined with augmented reality on mobile devices providing effective multi-touch collaborative working environments. The contributions of the presented work are the definition of a 2D multi touch composite table top, a 3D interaction style for portable multi-touch devices, the definition of a collaborative interaction environment and an evaluation procedure of 3D interaction techniques on multi-touch devices.

3.3. Methodology for multi-device 3D touch interaction

The aim of this chapter is to provide a solution to the problem of achieving more natural interaction mechanisms on portable devices through providing multi-touch collaborative interfaces with different 3D interaction techniques, using multi-touch portable devices (such as mobile smart phones and tablet PCs). To achieve that objective, two implementation approaches were evaluated: an approach based on the use of inertial sensors (such as gyroscopes, accelerometers and compass) to obtain user movement and provide 3D interaction; and an approach based on augmented reality using the integrated cameras of the devices to provide tracking, 3D positioning and interaction. The details of these methodologies are presented in the following subsections.

3.3.1. Inertial sensors' collaborative interaction approach.

In this section, the architecture based on inertial sensors to create multi touch collaborative interfaces focused on 3D interaction is presented. The proposed architecture will overcome the size and cost disadvantages of current large multi-touch displays by using interactive software and tablet PCs to create multi touch table tops of variable size and shape. By using a number of "connected" tablet PCs (or mobile devices) associated by their relative position, a virtual composite multi-touch table is created allowing interactions on a delimited workspace area. These devices have multi-touch capabilities, "3D position-awareness" and a networking system that can be used to achieve this task. Interaction with the 3D environment will be handled through the touch features of the device and a system developed to integrate movement detection based on the use of hardware components of the multi-touch portable device. This architecture will be used to share those interactions with the rest of the users in the network, to process common elements and to facilitate elements exchange between neighbour devices.

The device position is taken from the appropriate use of data provided by the accelerometers and the gyroscopes of the tablets able to provide the orientation and the displacement of the device to generate 3D interactions.

When the user starts the system, the tablet is initially placed in the "centre" of the virtual layer. The centre can be at any place and will correspond to the initial position of the tablet. Once we initialise the system, the centre of the whole "table" will be regarded at the same starting position and it will be used as a reference for the entire adjacent tablet PCs. After that process, the tablet can be "moved" around the virtual layer, showing different parts of the background, independently of its orientation. This is due to its "position-awareness" and movement provided by the accelerometer. A star network topology is used and the first tablet operates as a server for the remaining tablets that joined the table (the tablets can communicate via Bluetooth or by using a local wireless network). The second tablet gets the position information of the initial devices, using the data provided by the inertial devices and has to start from the same centre of the virtual surface. Since the second tablet has the information of the centre location both tablets will be linked as two parts of the whole array creating an "ad-hoc network", thus creating a fully functional multi touch table top. This process can be seen in Figure 3.1 (the overall representation and the position of the tablet PCs can be altered). The definition of the collaboration between tablets is going to depend of the context of the application, such as the passing of graphic elements between the tablets that are part of consecutive sections of the virtual interactive surface.

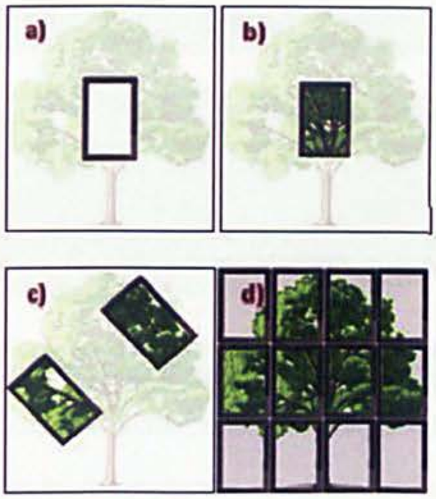


Figure 3.1: Proposed approach a) Start, b) Initial calibration, c) Multiple tablets, d) Full "table top" layer.

The proposed 2D approach is extended to support not only 2D interactions but also to incorporate also 3D environments, where our virtual layer is now a 3D scene containing models as the one shown in Figure 3.2.

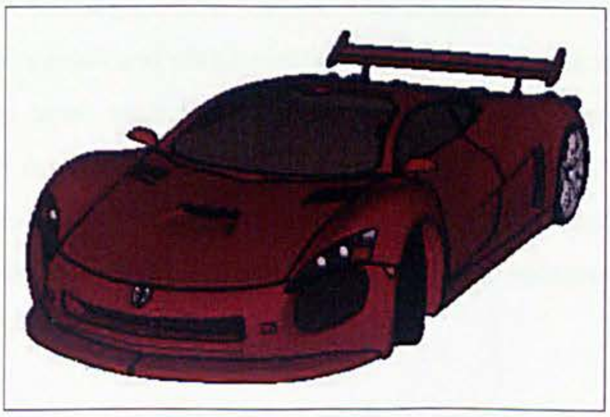


Figure 3.2: 3D model for calibration.

The system starts, again with a random initial view of the scene, such as front, side, top-down etc. The initial location and viewing direction will be regarded as the centre of the coordinate system and all subsequent positions will be estimated relative to that. The initial view of the 3D model in our scene will be similar to the 3D image in Figure 3.3; therefore the user needs to place the tablet on the top of the surface pointing downwards, above the surface that is going to be used to "place" the 3D model. The 3D model is displayed centred of the device' screen, generating a top view of the model.

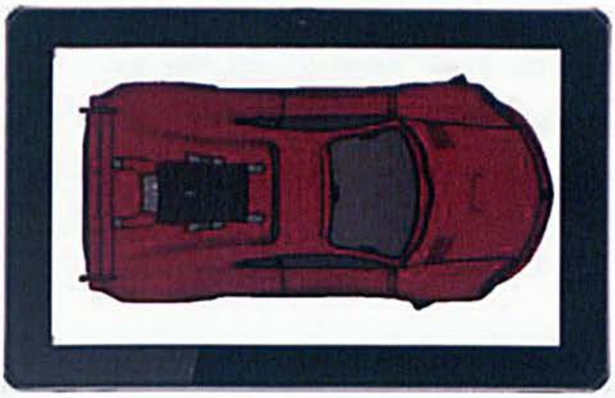


Figure 3.3: Start position for the system (top-down view).

Once the system is initialised, the user can freely move around the model, getting multiple viewpoints of the model and the 3D scene. This is possible due to the 3-dimensional self-awareness of the tablets (based on the measurements from the accelerometers and gyroscopes, which detect every change to the relative position of the tablet from the starting point). The user can choose different views by moving around the initial location and obtain results such as those in Figure 3.4. In order to make this process more understandable we could regard an example from FTV, where the viewers move around a virtual area e.g. football pitch selecting different points of view based on their personal preference and the actual events during the game. The views of the 3D model change according the movements, regarding the original position of the tablet pc.

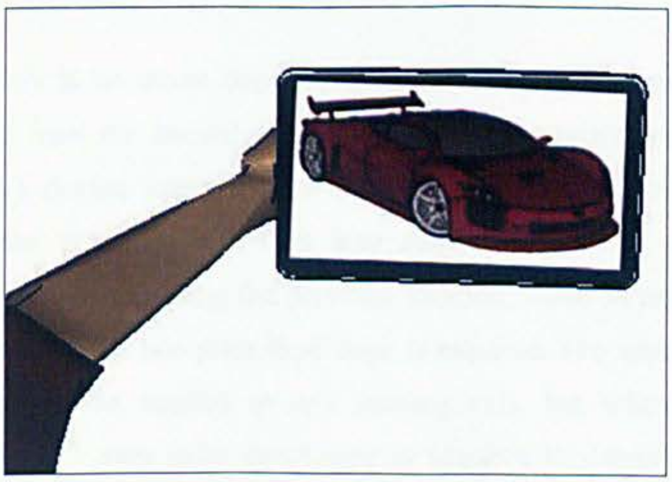


Figure 3.4: User view of the 3D model in the tablet PC.

After the setup of the first tablet and upon obtaining the relative position of the model, other users can join the "network" in a similar way, transferring information among the tablets regarding their relative positions and directions. As a result each user will have a different point of view of the model and a different perspective, (see Figure 3.5).

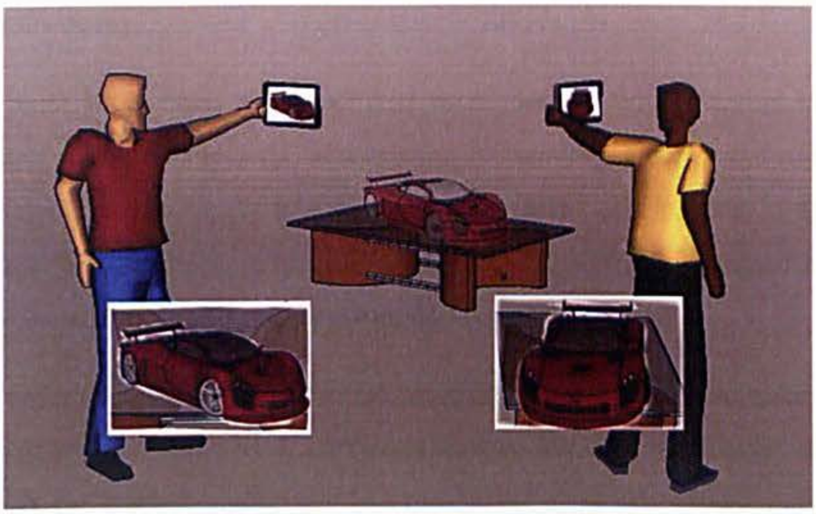


Figure 3.5: Multiple users with different point of views of the 3D model.

3.3.1.1. Implementation analysis

The system implementation was divided in two phases: the first phase evaluates the data provided by the inertial sensors of the portable devices (mobile smart phone and tablets) while functionality and system response are tested.

Since there is no direct conversion between acceleration and position, the signals obtained from the inertial sensors require further processing. Therefore to obtain position a double integral must be applied to the input signal to convert accelerations into velocities and then into displacement. This allows velocity information to be obtained using the previous location, while in case of the Verlet integrator the state of the *two* prior time steps is required. The approach utilised in this experiment can be applied to any sensing axis, but when positioning is implemented in all 3 axes extra processing is required to consider the effect of earth's gravity (Tansakenen et al., 2013).

To elaborate further; we use the readings from a digital accelerometer, gyroscope and the magnetometer integrated in a mobile or tablet device, and then combine them to form a rudimentary dead reckoning system that compensates for gravity. Dead reckoning is the process of estimating the current position by using a pre-determined start point, and then updating the device's position estimate through knowledge of its speed over time, and the direction the movement has been recorded. Each new estimate is calculated from old estimates; therefore error accumulation problems may occur.

In most tablet devices a triple axis accelerometer with a digital interface is available and we can sample the acceleration by sending a command over a serial digital protocol, which returns a number relating to the acceleration in terms of g-force. This is opposed to an analogue accelerometer, which is sampled by reading the voltage from its pins that is proportional to the acceleration.

In order to obtain the position and minimize the accumulated error, we apply the following process based on the previous section. Initially, the accelerometer is set to sample at the highest resolution and the fastest data rate. Since the system is initialised, the device can continuously provide the updated acceleration values at the specified data rate.

In most of the cases accelerometers may have offset errors; therefore calibration is required to remove this bias. Thus, a certain number of calibration readings are taken to estimate the bias, before we start our displacement calculations. By subtracting this bias from all the subsequent readings, we will have ensured that this offset is reduced.

$$ac_k = ac_k^{in} - b_k \quad with\, k \in \{x, y, z\} \qquad (1)$$

where

$$b_k = \frac{1}{N}\sum_{t=1}^{N} ac_{k_t}^{in}$$

with ac_k^{in} to represent the acquired data from the accelerometer in each axis, N is the number of data samples captured, ac_k is the unbiased acceleration and b_k is the estimated bias.

To obtain direction of the movement in 2D space the acceleration inputs are combined. As the accelerometer information is given in terms of *x*, *y*, *z*, we can select the axis and combine them to represent the movement. In order to incorporate 3D movement we have to take into account the third axis and compensate for the gravity. This is due to the accelerometer being subject to dynamic and static (gravity) accelerations. Since we are interested in measuring dynamic accelerations rather than gravity, a simple approach was introduced assuming that the device has an accelerometer, gyroscope and magnetometer. Using the readings from the gyroscope smoothed with that of the other two sensors a quaternion representing the orientation of the Earth frame with respect to the Sensor frame can be evaluated as a result of sensor fusion. Consequently, we can compensate for the gravity and obtain the dynamic acceleration using the following equations.

$$gc_x = ac_x - 2(q_x q_z - q_r q_y)$$

$$gc_y = ac_y - 2(q_r q_x - q_y q_z)$$

$$gc_z = ac_z - q_r^2 - q_x^2 - q_y^2 + q_z^2 \quad (2)$$

Where each gc represents the calculated acceleration in each axis, given $q = q_r + q_x i + q_y j + q_z k$ that represents the orientation quaternion provided by the sensor fusion data, $ac = ac_r + ac_x i + ac_y j + ac_z k$, which corresponds to the readings coming from the accelerometer and $gc = gc_r + gc_x i + gc_y j + gc_z k$ is a 3D vector that represents the dynamic acceleration. In this approach using quaternions, we compute the expected direction of gravity and then subtract that from the readings of the accelerometer. The orientation of the device is also given by the previous described equations.

An update method is considered to sample the accelerometer and perform the calculations for position estimation. Accelerometers are quite noisy, and running averaging is applied to reduce the error introduced by the noise without introducing noticeable delays on the frame rate.

Additionally, two thresholds are selected experimentally (using previous data from the devices) to indicate the range of the invalid values of the accelerometer. If an averaged signal fell within these limits we assume them as noise and the acceleration is set to zero.

In order to obtain the position, we regard the velocity as the derivative of the position and the acceleration is the derivative of the velocity. Thus integrating the values we have

$$v = \int(\vec{a})dt \text{ and } s = \int(\vec{v})dt = \int(\int(\vec{a})dt)dt \quad (3)$$

where

$$a = \frac{d\vec{v}}{dt} = \frac{d(\overrightarrow{ds})}{dt^2} \text{ and } v = \frac{d\vec{s}}{dt}.$$

We can simplify this approach by introducing a numerical solution and in this case we obtain:

$$v(n) = v(n-1) + a(n)dt \qquad (4)$$

$$s(n) = s(n-1) + v(n)dt + \frac{1}{2}a(n)dt^2 \qquad (5)$$

By applying this to the acceleration readings, and then again to the velocity calculations, we can obtain an estimate of position.

Another issue that needs to be addressed is the estimation of the end of the movement. If we move our accelerometer from a stationary point for a certain distance and then bring it to rest again, we should get an equal and opposite amount of acceleration in both directions. If we don't read an equal amount, our velocity will remain constant at a certain number. A heuristic approach is utilized and the number of zero readings of acceleration in a row is calculated. If that number is above a predefined threshold the velocity is reduced gradually to zero.

The view of the object is generated using the original view of the object from the start position and later on, giving the movements of the device, the subsequent transformations of coordinates are applied to the model.

The experimental results showed the use of inertial sensors by themselves to generate collaborative 3D interaction is not reliable enough, since the amount of the accumulated error is significantly high for such applications that require more precision on the device 3D position estimation. These results will be analysed further in section 3.5.

3.3.2. Augmented reality collaborative approach

The approach presented in this section is based on the use of computer vision algorithms to generate a 3D augmented reality interaction environment for collaborative work, implemented on portable devices. The proposed approach aims to generate 3D movement-based interaction able to function in a network of collaborative systems, based on 3D position awareness to overcome the issues described in the previous approach. Instead of using inertial sensors, the camera integrated in the portable devices is used to calculate the relative position of the devices thus generating an augmented reality movement-based interactive 3D system.

In this approach, relative position of the devices is calculated using a target image for calibration that provides characteristic points clear enough to determine the position of the device in 3D space. Figure 3.6 shows the calibration image used in our approach.

Figure 3.6: Calibration image to generate the augmented reality collaborative environment.

The system starts when the users aim to the calibration image, which must be placed on a plain surface where the interaction and the virtual environment will be located. Given the relative position of the users' devices with respect to the target (captured by the integrated cameras); the virtual 3D scene will be shown in the portable device's screen as is shown in Figure 3.7.

Figure 3.7: Visualisation of the 3D model/scene using a calibration image to provide interaction.

After the initialisation process, the user can move around the observed model, always aiming to the target to keep the position of the model. The 3D scene and all the changes on the observed models are calculated and displayed based on the relative position of the target to the portable device (the implementation details are discussed in section 3.1.2.1).

The interaction with several users is implemented in a similar way to the approach based on inertial sensors. However, the 3D model coordinates are obtained from the individual detection of the target by each device. In that way, the users can move around the model (always aiming the target) and share interaction information, meanwhile the visualisation process is generated in the individual devices.

3.3.2.1. Implementation analysis

The implementation of the 3D approach, based on augmented reality vision-based algorithms is traditionally performed using two basic phases: estimation of the transformation matrix and tracking of the target image.

The estimation of the transformation matrix is necessary to calculate the relative position of the camera (integrated to the portable device) and the calibration image (Kato and Billinghurst, 1999). This matrix will provide the necessary information to establish the relationship between the device's camera coordinates and the real world. Figure 3.8 shows an example of the different coordinate systems involved.

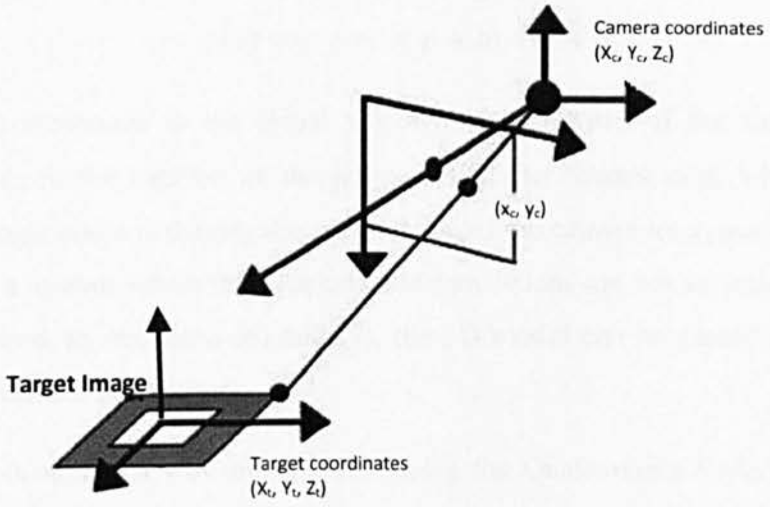


Figure 3.8: Coordinate systems involved in the vision based augmented reality model.

Figure 3.8 presents the coordinate systems involved in the vision-based 3D approach, where X_c, Y_c and Z_c correspond to the axes related to the camera's coordinate system, and X_t, Y_t and Z_t correspond to the axes of to the target image plane. The transformation from the camera coordinate system to the target image coordinate system is given by:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = T_{cm} \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} \qquad (6)$$

where T_{cm} corresponds to the transformation matrix given the characteristics points (specific points of the image, such as corners, convexity points, junctions, etc.) of the calibration image and the coordinates related to the real world. This transformation matrix is calculated in different ways depending on the algorithm used. This transformation allows placing the virtual 3D object over the target space. After that, another process has to be performed to track the camera movements and its 3D position.

The process of tracking the motion of the camera is performed using a motion model for rigid body motion (Koller et al., 1997), widely used for augmented reality applications with six degrees of freedom and camera motions of constant velocities. The motion equation for a point p in our virtual 3D object in a time t is given by:

$$p(t) = v + \omega \times p + at \qquad (7)$$

where v corresponds to the initial velocity of movement of the camera, $\omega \times p$ corresponds to the rotation of the point around the camera axes, where ω is the rotation angle and a is the angular acceleration of the camera movement. This model describes a system where the rotations and translations are not affected by external forces. Based on equations (6) and (7), the 3D model can be placed and followed along the camera movements.

This approach was implemented using the Qualcomm's Vuforia framework (Kasahara et al., 2012; Heun et al., 2013), designed to be used in applications based augmented reality on portable and mobile devices with touch capabilities (resolving the estimation of the transformation matrix and the tracking of the target image obtaining the 3D position of each user.

The definition of the interface for collaborative experiments is presented in the following section.

3.4. Collaborative interface definition

The interface to evaluate our interaction model was designed to support collaborative interaction under a 3D interface, based on touch interaction. The devices used for this were tablet PCs able to support visualisation and manipulation of 3D interfaces.

The interaction defined for this 3D touch based interface was focused on two basic aspects:

- Movements: The system provides the capability to move around a 3D scene, allowing the user to see the interactive 3D models from any possible point of view. The method to provide this interaction was defined as movement based (moving the device to see different points of the model) or touch based (with a panel to provide the movement in different directions).
- Actions: The actions over the 3D model must be based on touch interactions and should be able to reflect a change in the 3D model in a specific way. These actions also should reflect interactions of multiple users over the same 3D model, without interrupting the capability of each user to have multiple views of the 3D scene.

To achieve that, a cooperative game based on a 3D interface was designed. The idea of this evaluation game is that the users should remove sections of a 3D cube to reveal an inner 3D shape, simulating the sculpting process. The user must be capable of seeing the cube from different angles (movement aspect) and remove the sections of the cube (grey cubes) to reveal the inner 3D shape, using taps over the screen (action aspect). Each time the user removes successfully a section, the score increases. Since the objective of the game is to reveal a 3D structure inside of the cube, each time the user hits one of the cube elements corresponding to the model (represented by blue cube elements), an error counter increases.

The 3D cube model was one as the presented on Figure 3.9.

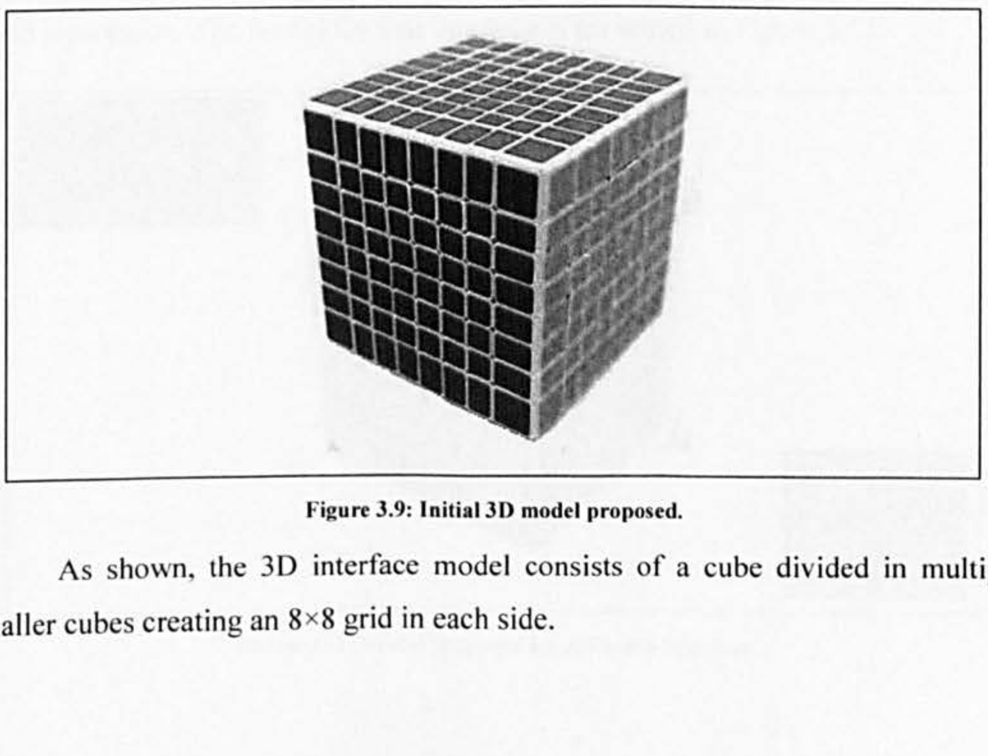


Figure 3.9: Initial 3D model proposed.

As shown, the 3D interface model consists of a cube divided in multiple smaller cubes creating an 8×8 grid in each side.

The general interface also presents the information of the correct hits and the wrong ones. A model of this interface can be seen on Figure 3.10.

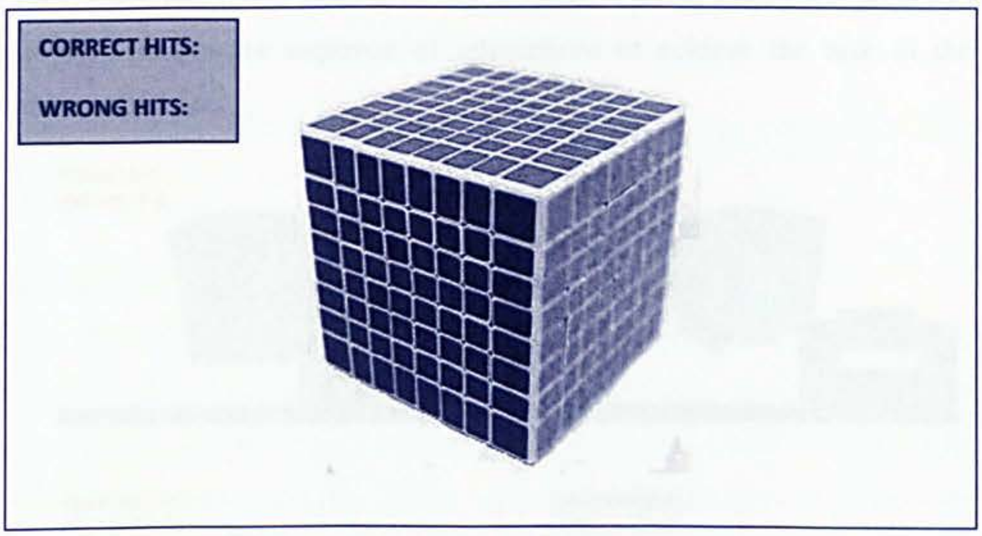


Figure 3.10: 3D model proposed plus the score information area.

The score information is placed at the top left corner to provide information for users without interfering with the interaction.

The evaluation process of the interaction interface was divided in two stages: the single user stage to evaluate interaction and interface features; and the collaborative stage, to evaluate collaborative interaction mechanisms.

Since the movements in the 2D touch interface are not based on the device's movements, this interface must provide a mechanism to rotate the cube, based on touch interaction. The model for that interface is presented in Figure 3.11.

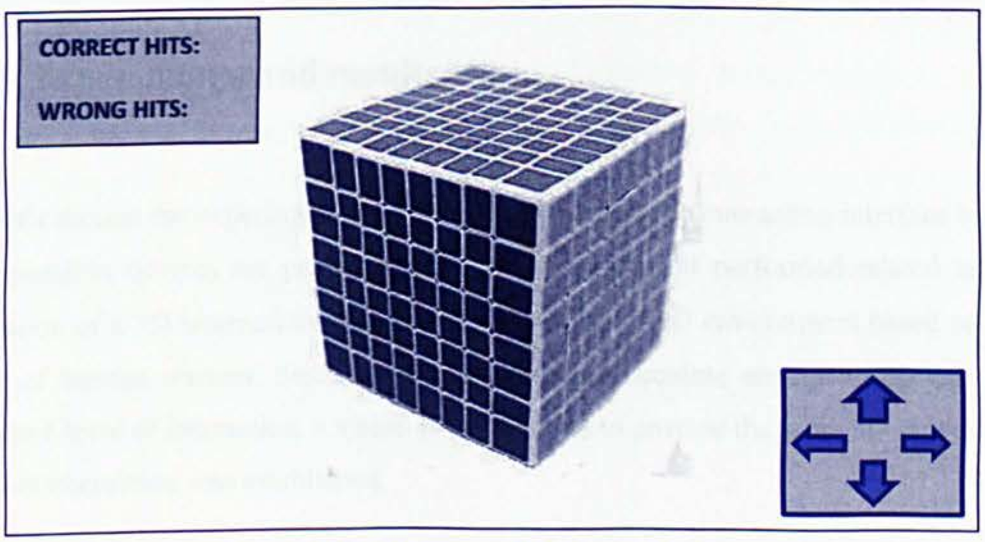


Figure 3.11: Model proposed for the touch interface.

The rotation panel is placed on the lower right corner of the interaction screen, as can be seen in Figure 3.11. This rotation mechanism is necessary to allow the visualisation of all the sides of the cube and the corresponding sub sections of it. Figure 3.12 shows the sequence of interactions to achieve the task in the 2D interaction approach.

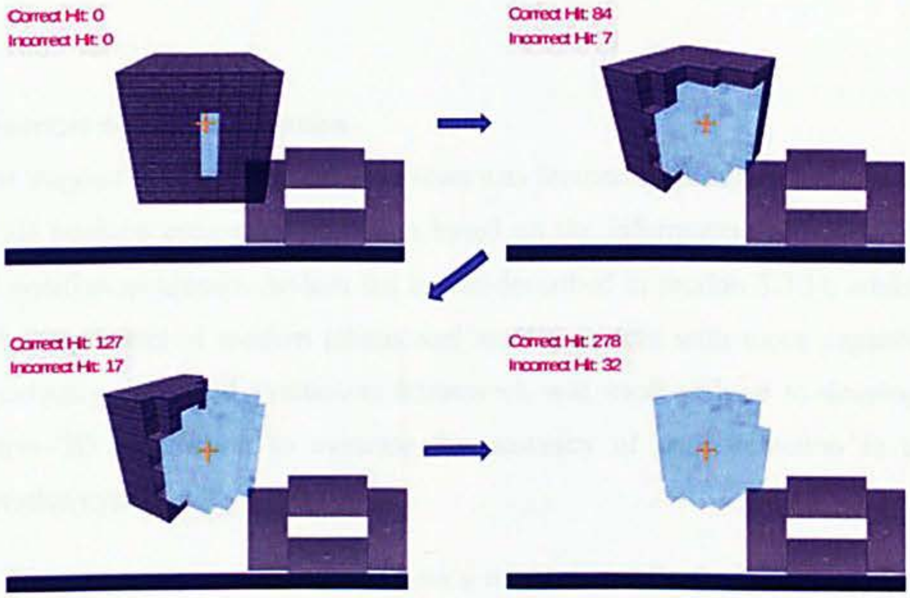


Figure 3.12: Interaction sequence in the 2D interaction approach.

The experiments to achieve this interface and the results of these experiments are presented in the following section

3.5. Experiments and results

In this section the experiments to create and evaluate a 3D interaction interface based on portable devices are presented. The first experiment performed related to the creation of a 3D interaction system operating in a real 3D environment based on the use of inertial sensors. Since the results were not accurate enough to provide the desired level of interaction, a vision based method to provide the same 3D movement based interaction was established.

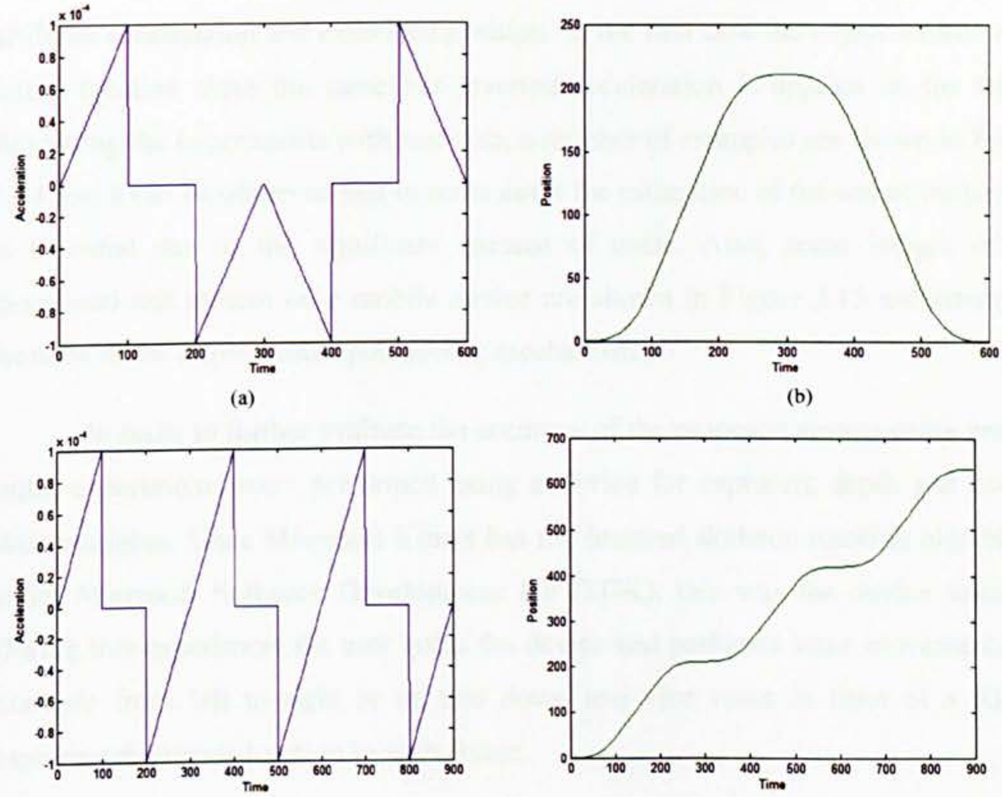
As it was necessary to compare the capabilities of the interaction model presented versus a typical 2D interaction, an alternative touch interface capable to offer similar features was developed and tested against our 3D interaction approach in two specific scenarios: single and collaborative interaction.

The results of the experiments previously described are presented as follows.

3.5.1. Inertial devices' evaluation

The first stage of the experimental procedure was focused on measuring the accuracy of the 3D position estimation approach based on the information provided by the inertial position awareness devices (as it was described in section 3.3.1), which are internal components of modern tablets and mobile devices with touch capabilities. The previously described evaluation framework was used as base to develop the interactive 3D application to evaluate the accuracy of the interaction in a 3D collaborative system.

The test system was developed on a modern mobile device, equipped with the required sensors and running current dual core 1.5 GHz processor with 1GB RAM. The developed system acquires the measurements from the accelerometer and applies the integration process described in section 3.3.1.1. Initially the system estimates if there is a bias on the accelerometer by introducing a calibration step during which the user places the device in its initial position without moving for a set amount of time (less than few seconds). During that period the incoming data are accumulated and the mean value obtained is used to remove the bias from all the upcoming acquisitions. Furthermore, since the incoming data are prone to noise, moving filters were designed and implemented to reduce the error during the integration process. The moving filters due to their nature introduce a delay of less than 1/4th of a second, which could be adjusted to fulfil the accuracy requirements of the application. The sampling rate from the accelerometer is up to 60 samples per frame, with the application running at 40 frames per second, this being a considerable number of samples when scaled down to the 24/25/30fps of traditional broadcasts, thus affording us longer moving filters.



(a) (b)

(c) (d)

Figure 3.13: Artificial examples with the acceleration (a), (c) and the estimated position versus time (b), (d).

The initial position of the device is regarded as the centre of the coordinate system and the integration is then applied using the Verlet Integrator. A threshold was defined experimentally to allow the estimated position to halt if accelerometer inputs to the Verlet Integrator were zero for a sufficient number of times. This is due to the device assuming it is still in motion caused by signal noise. During these rest stages, the system could recalibrate in order to reduce error propagation.

In our experiments two moving filters were used to reduce the noise, the moving average and the moving maximum. Both of them are approximately 20 samples in length and in case of the moving maximum, the maximum acceleration value of the current window is selected while in the case of the moving average the mean value is used.

From the experiments performed, we can see in Figure 3.13 two cases of artificial acceleration and estimated position. In the first case the object returns to its initial location since the same but inverted acceleration is applied on the tablet. Regarding the experiments with real data, a number of examples are shown in Figure 3.14 and it can be observed that in some cases the estimation of the end of movement is essential due to the significant amount of noise. Also, some images of the developed test system on a mobile device are shown in Figure 3.15 indicating the features of the implemented positioning mechanism.

In order to further evaluate the accuracy of the proposed system using ground truth, experiments were performed using a device for capturing depth and human skeleton joins. Since Microsoft Kinect has the featured skeleton tracking algorithms in its Microsoft Software Development Kit (SDK), this was the device selected. During this experiment the user holds the device and performs some movements for example from left to right or up and down and vice versa in front of a Kinect capturing the hands location in each frame.

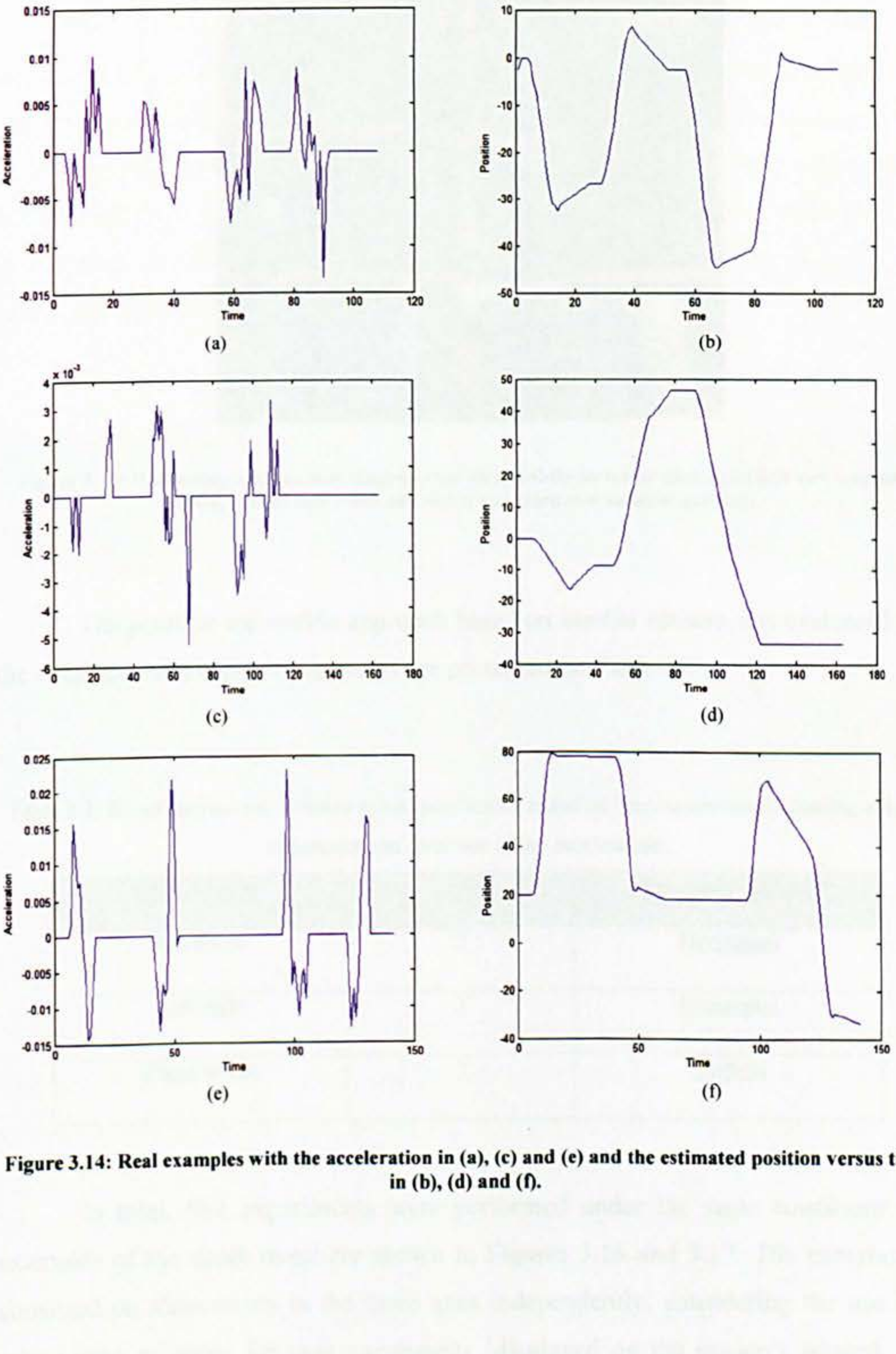


Figure 3.14: Real examples with the acceleration in (a), (c) and (e) and the estimated position versus time in (b), (d) and (f).

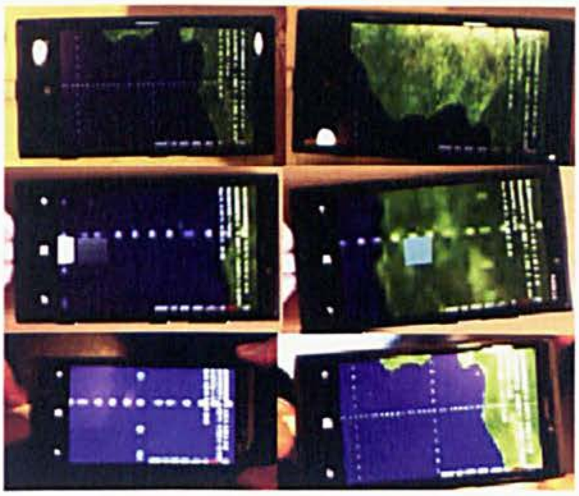


Figure 3.15: Positioning mechanisms implemented on a mobile device on each axis (first row – up and down, second row – left and right; and third row zoom in and out).

The position estimation approach based on inertial sensors was evaluated and the axes involved in the experiments are presented on Table 3.1.

Table 3.1: Set of movements selected to test positioning based on inertial devices, indicating axis of movement and direction of the acceleration.

Movement	***Axis***	***Acceleration direction***
Up-down	*Z*	Horizontal
Left-right	*X*	Horizontal
Zoom in-out	*Y*	Vertical

In total, five experiments were performed under the same conditions and examples of the depth maps are shown in Figures 3.16 and 3.17. The experiments consisted on movements in the three axes independently, considering the use of a map image as guide for user movements (displayed on the device's screen). The movements evaluated considered two sets of horizontal movements ("up-down" on the map for *Z-axis* and "left-right" in the map for *X-axis*) and one set of vertical movements ("zoom in-out" in the *Y-axis*). In order to obtain the location of the device used as ground truth, the average 3D position in between the two hands was calculated using Kinect and the skeleton tracking features previously mentioned. The first three experiments contain mainly horizontal movements of the device. The sequence of movements considered in the horizontal tests were movements from the start position to the left, from the start to the right and finally, from the start position to front (trying to avoid any vertical displacements). The last two tests contain mainly vertical movements with the gravity effect more noticeable, displacing the device from the start position to an upper or lower position (trying to avoid any horizontal displacements).

Figure 3.16: Example of the depth map indicating the user moving the device left and right.

Figure 3.17: Example of the depth map indicating the user moving the device from top to bottom.

In Figures 3.18, 3.19 and 3.20 results are presented from three experiments with horizontal movement in the first two (*Z* and *X* axes movements) and vertical in the last one (*Y* axis). In each figure, the acquired acceleration, position and error values for the different set of movements are plotted versus the time. Also the filtered versions using both the moving maximum and the moving average filter are shown. Also the ground truth location obtained by the Kinect and the estimated positions for both filters using the integration approach are shown.

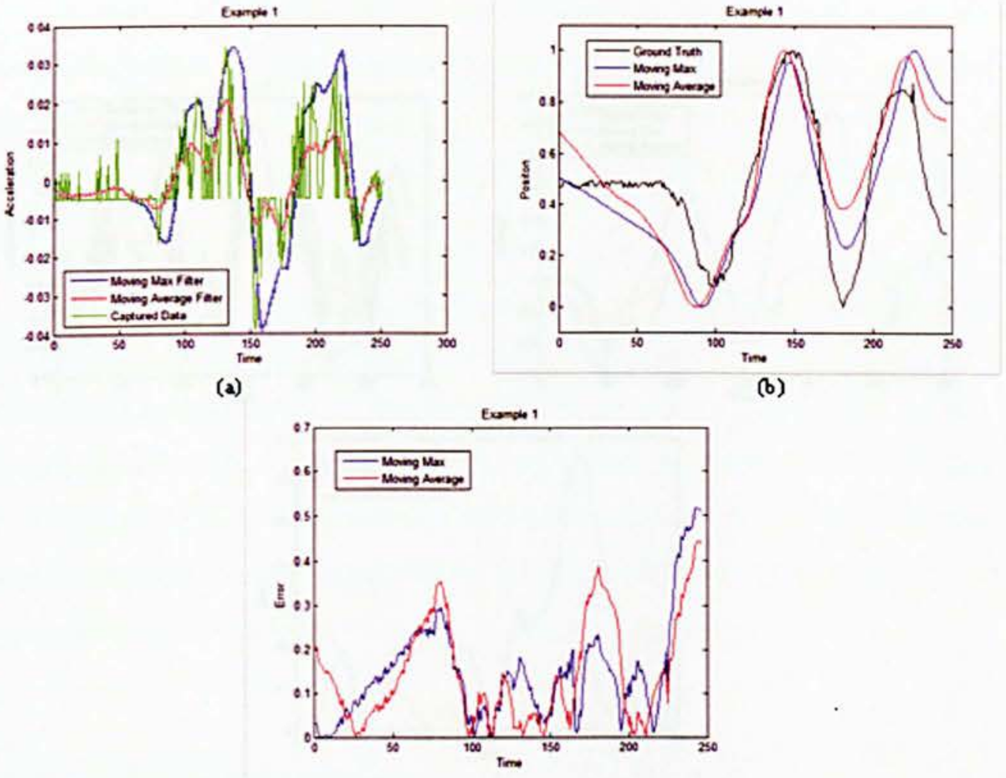


Figure 3.18: Example 1 of horizontal movement a) acceleration before and after the moving filtering b) the estimated position using integration and c) the obtained error for both filters. The error is show in terms of the ratio to the ground truth.

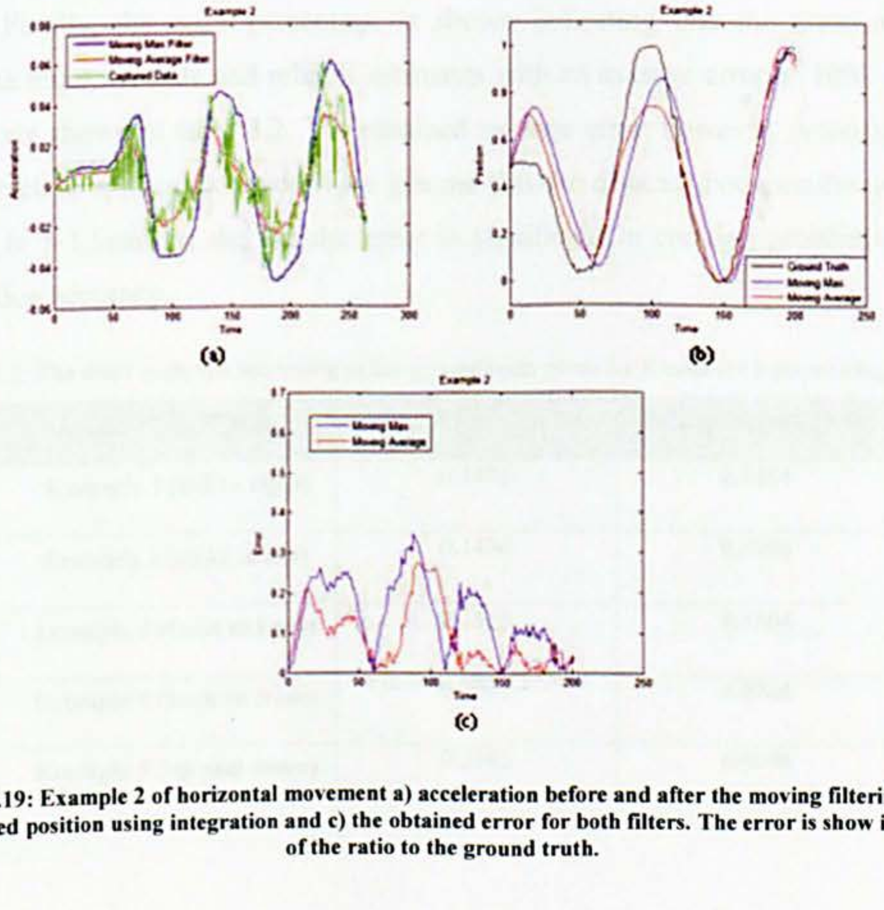


Figure 3.19: Example 2 of horizontal movement a) acceleration before and after the moving filtering b) the estimated position using integration and c) the obtained error for both filters. The error is show in terms of the ratio to the ground truth.

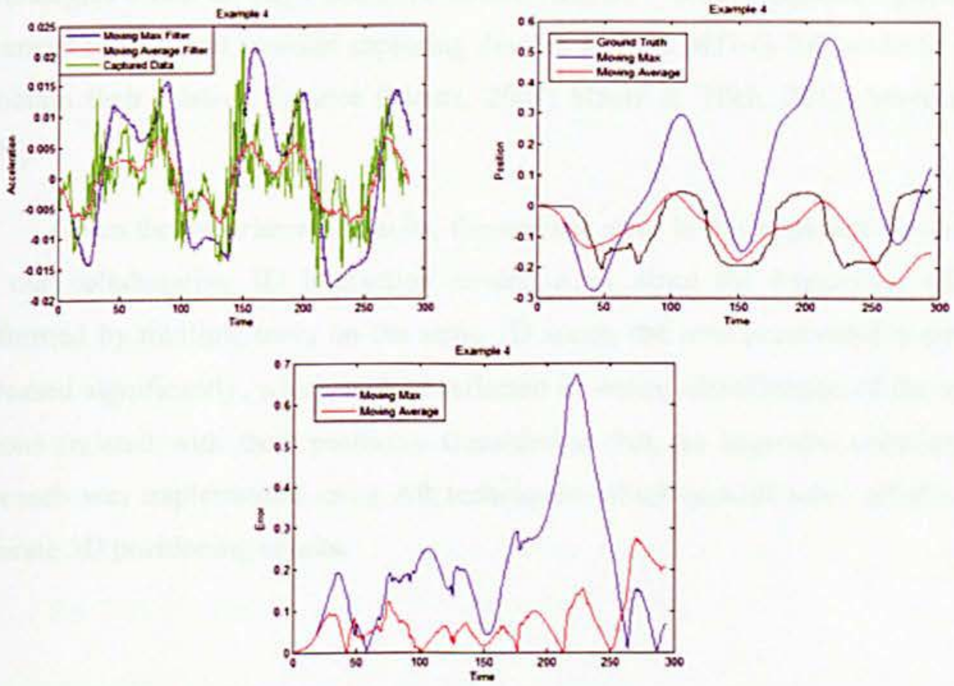


Figure 3.20: Example 4 of vertical movement a) acceleration before and after the moving filtering b) the estimated position using integration and c) the obtained error for both filters. The error is show in terms of the ratio to the ground truth.

Finally, the error percentage is shown indicating that the mean average provides more accurate and reliable estimates with an average error of 10%. All the results are shown in table 3.2. The obtained average error, however, is too high for an interactive application since if we assume that the distance between the user and Kinect is 1-1.5meters, the angular error is significant in creating problems to the interaction accuracy.

Table 3.2: The error is shown according to the ground truth given by Kinect for both moving filters.

Movement Examples	***Moving Max (ratio)***	***Moving Average (ratio)***
Example 1 (left to right)	0.1493	**0.1464**
Example 2 (right to left)	0.1456	**0.0906**
Example 3 (front to back)	0.1898	**0.1304**
Example 4 (back to front)	0.2011	**0.0745**
Example 5 (up and down)	0.2162	**0.0946**

In order to improve the results obtained in these experiments, other technologies based on High Sensitive GNSS, WLAN / WiFi, Magnetic Systems or advanced motion and position capturing devices such as MTi-G-700 could be used to obtain their relative distance (Mautz, 2009; Mautz & Tilch, 2011; Maye et al, 2006).

Given the experimental results, the average error in this approach is too high for our collaborative 3D interaction model. Also, since the interaction will be performed by multiple users on the same 3D scene, the total accumulative error is increased significantly, which will be reflected in wrong identification of the users' actions (related with their position). Considering that, an improved collaborative approach was implemented using AR techniques, which provide more reliable and accurate 3D positioning results.

3.5.2. User's interaction evaluation

The experiment to evaluate user interaction performance is presented in this section. This evaluation process considers a qualitative and quantitative assessment. The interface presented in section 3.4 was used to test the performance of the users in two basic scenarios: single and collaborative interaction. Also, each of these evaluation methods has two stages: interaction based on 3D positioning using AR and interaction using a simple 2D on screen touch interaction interface. The main objective of this task is to compare 2D touch interfaces with the proposed 3D positioning interface and determinate if there are important advantages provided by the proposed method, especially in collaborative assignments. The defined task is to remove all the surface cubic components (278 grey cubes in total) to reveal the inner 3D shape. This removal process must be done without "hitting" the components of the 3D inner shape (blue cubes, at the surface or inside the model) in a simulation of a single and collaborative sculpting process. Two interaction techniques (2D touch based and 3D movement based approach) must be performed to compare their results and evaluation by the users. The interface was selected because it provides two key elements to validate the model: use of touch interaction and use of movement to visualise a 3D model in a real environment.

In order to provide the necessary interaction mechanisms described in section 3.4 for single user and collaborative experiments, the following commands were defined for the 3D interaction approach:

- **Cube rotation:** The cube rotation is achieved by moving/positioning the device around the 3D scene, providing the visualisation of the interaction area from all possible sides. The rotation is necessary to provide the interaction with all the sides of the 3D model.
- **Aiming and elimination of cube's components:** To remove the grey components of the cube 3D model, it is necessary to tap on the screen. To avoid the occlusion of the model by the tap gesture, the interface has an aiming indicator (a cross in the centre of the screen) to target the cubes that should be removed after using the rotation interaction provided. A model of the interface plus the aiming system is shown in Figure 3.21.

The commands defined for the 2D touch interaction (2D approach) are:

- **Cube rotation:** The cube rotation is provided by 2D movements on a specified touch area of the tablet's screen, as it was described previously. This 2D touch area provides 4 directions of movements (up, down, left and right), which allow to rotate the cube in any 3D direction and to visualise any of the cube's faces.
- **Aiming and elimination of cube's components:** The action to remove grey cubes is the same as in the 3D approach, using an aiming indicator (similar to the one presented in Figure 3.21) and tapping on the screen. The aiming system allows the user to remove the cubes by tapping anywhere on the screen, except the areas used to display the score, the model and the rotation panel.

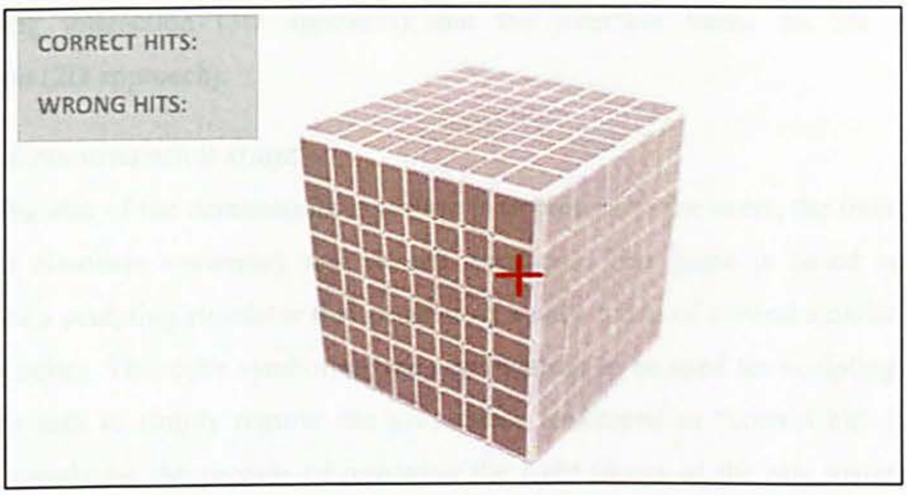


Figure 3.21: Model proposed for the touch interface with aim system.

The 2D and 3D interfaces under a collaborative mode are evaluated using the same approaches but the users work in pairs in cooperation to achieve the task previously described (i.e. remove the 278 cubes and reveal the hidden 3D model), but in this case for each user their individual amount of correct hits is counted allowing a competitive element in this task. Also the wrong hits are being counted individually which permits to evaluate the accuracy of the individual user's interaction. The users have direct feedback to their counterpart actions in their individual screens. The feedback between users is achieved directly by the networked connection between the portable devices, as was explained in section 3.3.2.

To perform this experiment, the procedure was explained to each user. The following sections clarify the steps followed during the experimental procedure:

3.5.2.1. Experiment's presentation and its objectives

The objective of this section is to inform the users about the objective of the experiment: comparison of 2D and 3D interaction interfaces evaluating their advantages and disadvantages. This experiment also aims to collect information about the level of convenience of the 3D interaction in a 3D environment using portable devices. The first stage of this test is related to single-user interaction and the second, to collaborative interaction (two users working for the same objective). Both stages are divided in two parts: the interface based on the 3D movement and positioning interaction (3D approach) and the interface based on 2D touch interaction (2D approach).

3.5.2.2. Demonstration stage

The aim of the demonstration section is to present to the users, the interfaces and their elements answering any related questions. The game is based on the concept of a sculpting simulator that consists of a cube made of several smaller grey and blue cubes. This cube symbolises the raw material to be used for sculpting with the user's task to simply remove the grey cubes (indicated as "correct hit" in the screen), simulating the process of removing the right pieces of the raw material to reveal the sculpture's shape (the 3D model inside of the cube). If the user tries to remove the blue cubes, that action increases the error counter (indicated as "incorrect hit" on the screen). The objective of the game is to remove all of the grey cubes in the interface to reveal the blue 3D model, which is the final model of the sculpting process and obtain the desired figure (statue) from a block of raw material.

The main aspects to explain are related to the game procedure. The instructions for the game are:

SINGLE PLAYER- 3D movement interface (3D approach):

- To remove cubes in this mode, the user must move around the object and aim towards the cube she/he wants to destroy. Once the desired cube is aimed, the user just needs to tap the screen.

SINGLE PLAYER: 2D touch based Interface (2D approach):

- As with the 3D approach the game is the same but the difference lies in the control mode.

To remove the grey cubes in this mode, the user must use the 2D touch area in the interface to rotate the cube (based on the 2D movement panel) and aim at the cube to be removed. Once the cube to be removed is aimed, the user just needs to tap on the screen (except the area assigned to rotate the cube). Also, rotation of the 3D scene and removal of grey cube elements cannot be performed at the same time.

MULTIPLAYER (3D approach)

- In this case, one of the users is going to be the host and the other the guest in order to establish a connection between them.

- Once the server is created, the guest user must join that server.
- In this case, the removal of the grey cubes becomes a cooperative task (both users interact over the same cube), but their hits and misses count individually.
- The interaction with the cube (movements and removal of grey components) is the same as in the single 3D approach case.

MULTIPLAYER (2D approach)

- In this case, one of the users is going to be the host and the other, the guest of the application
- Once the server is created, the guest user must join the server.
- In this case, the removal of the grey cubes is cooperative (both users interact over the same main cube), but their hits and misses are counted individually.
- The interaction with the cube (movements and destruction) is the same as in the single "2D approach" case.

3.5.2.3. Familiarise the subject with the interface

During the stage of familiarisation with the interface, the interaction mechanisms are presented to the users, allowing them to practice basic movements and the on screen features.

3.5.2.4. Subjects perform the available functionalities

The available functions (e.g. aim and remove grey cubes) are performed by the user, interacting with both interfaces (3D positioning and 2D touch approach) and trying to perform different actions (move around the 3D model and remove the grey areas to reveal the blue 3D shape).

3.5.2.5. Quantitative data collection

Once the users familiarise themselves with the environment and understand how to perform the available functions, then the full task that was initially defined is performed. First, the users work with the single user interface and then with the multiplayer (collaborative) one interacting with the other user. During the experiment the quantitative aspects to be evaluated are going to be time to perform the task, total correct hits and total misses.

3.5.2.6. Qualitative evaluation approach

After the completion of the task a questionnaire about the interaction mechanism's experience, evaluating and comparing the available interfaces is given to the users. The single approach is performed to evaluate qualitatively the interface itself; while the collaborative approach aims to evaluate the collaborative features (such as feedback from other user's actions, experience simultaneous working environments, functionality of collaborative interactions, etc.).

The model of questionnaire that was used is based on the questionnaires provided by IBM in their research about new interfaces on usability tests (Lewis, 1995). For our experiment, two questionnaires were presented: the first related to the single interaction and the second to the collaborative interaction interface.

The first questionnaire is separated into two main sections in order to evaluate the user experience with the interface, and is shown in table 3.3. In all questions the user provides a score from 1 to 5 to evaluate the interface according to the question, where 1 is the lowest score (extremely negative evaluation) and 5 is the highest (extremely positive evaluation).

Table 3.3: Usability questions for single interaction.

Section 1	Section 2 - How would you rate:
QS1: Was the interaction easy to understand?	QS4: The Interface?
QS2: Was it easy to manipulate?	QS5: The Performance?
QS3: Is the navigation system intuitive?	QS6: The functionality?
	QS7: The objective achieved?
	QS8: The user experience?
	QS9: The commands selected (i.e. tap, move around)?

The second questionnaire was created to obtain the feedback of the users after the collaborative interaction.

Table 3.4: Usability questions for collaborative interaction.

Questions
QC1: The feedback from the other user's actions?
QC2: The Performance?
QC3: The functionality?
QC4: The objective achieved?
QC5: The user experience?

The software used to develop the 3D interface for both approaches was based on the Qualcomm's Vuforia framework providing the augmented reality features (as was described on section 3.3.2.1), the Unity3D game engine and the C# programming language was used for general programming of the software.

The portable devices used in the testing process were tablet PCs. The hardware features of the tablets are a dual core processor, 2GB of RAM, quad-core GPU (for 3D graphics), a rear camera (necessary for the augmented reality interaction interface) of 5 megapixels and Wi-Fi internet connection (to provide the network connection for the collaborative experiments). The operating system used was Android 4.2.2, which provides the capabilities to support the software configuration previously presented.

The start of the system requires a calibration image. The calibration image needs to provide recognisable characteristic elements to generate the system's calibration and tracking. The image used on the real interaction environment is shown in Figure 3.22. This image was created to provide several graphic recognisable elements (such as square, junctions and letters) which can make the identification of the orientation of the image in the real world, easy and consequently allows to placing a 3D model in a specific position using the algorithms provided by Qualcomm's Vuforia SDK.

Figure 3.22: Calibration target in real interaction environment.

After the system identifies the pattern, the 3D interface is displayed on the screen (see Figure 3.23).

Figure 3.23: 3D interface on the real interaction environment.

In the case of the simple touch application (2D approach), the interface is similar to the one presented on Figure 3.11. The implemented interface can be seen on Figure 3.24

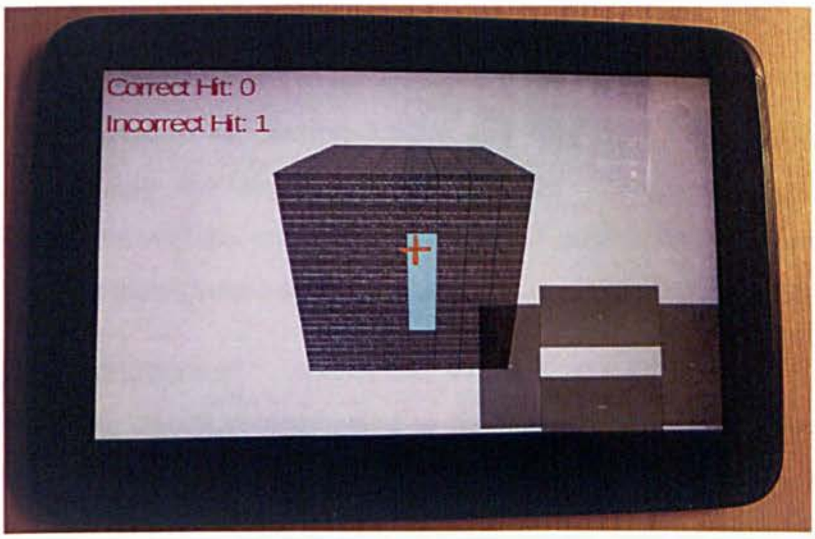


Figure 3.24: 2D interface on the real interaction environment.

In order to evaluate the proposed interfaces, the interaction experiments were performed using 10 subjects for the single interaction and 20 subjects for the collaborative interaction interfaces (i.e. 10 pairs). The range of ages of the subjects for both experiments varied between 20 and 50 years old, with an average age of 31 years old in both the single user and collaborative experiments. Their knowledge of touch interfaces varied from occasional to expert users. The results of the experiments are separated into two groups: qualitative and quantitative results, and analysed in the following sections.

The statistical validation of the data was performed using the Wilcoxon signed-rank test (Rosner et al., 2006). The Wilcoxon test is used in cases where the data cannot be considered normally distributed and there are paired values of a given amount of subjects, which is the case of our data (previous analysis performed to the data obtained shows it level of skewness is too high to be considered normally distributed). To obtain the statistical significance given an amount of experimental subjects or samples N, a statistical value called "*Wilcoxon statistical*" is obtained, which is lately used to obtain the p-value associated, which is the probability of obtaining a test statistic result at least as extreme or as close to the one that was actually observed (Goodman, 1999). If the p-value is less than 0.05, there is a strong assumption that the results will occur in a similar experiment.

3.5.2.7. Qualitative results

The following results are related to the feedback given by the users according to the questionnaires presented on section 3.5.2.6 and they will be presented in this subsection separately for single and collaborative interaction interfaces. The summarized results will be presented on terms of median and median absolute deviation to avoid the influence of outliers.

3.5.2.7.1 Single interaction

In this section, the results corresponding to the answers given by the users in the questionnaire of single interaction are analysed. The summary of these answers is shown on Table 3.6. The complementary information provided by users' feedback is summarised in the following tables and graphs. Table 3.5 shows the obtained median scores for the answered questions on the "3D approach" and "2D approach" (with the median absolute deviation shown in the brackets). The best results for each section are highlighted (bold values).

Table 3.5: Median values and median absolute deviation in each question for both single interfaces.

<table>
<tr><th>S 1</th><th colspan="2">QS1</th><th colspan="2">QS2</th><th colspan="2">QS3</th></tr>
<tr><td>Median 3D approach</td><td colspan="2">5.0 (0.0)</td><td colspan="2">4.5 (0.5)</td><td colspan="2">4.0 (0.5)</td></tr>
<tr><td>Median 2D approach</td><td colspan="2">4.5 (0.5)</td><td colspan="2">3.0 (1.0)</td><td colspan="2">4.0 (1.0)</td></tr>
<tr><th>S 2</th><th>QS4</th><th>QS5</th><th>QS6</th><th>QS7</th><th>QS8</th><th>QS9</th></tr>
<tr><td>Median 3D approach</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>5.0 (0.0)</td><td>5.0 (0.0)</td><td>4.0 (1.0)</td><td>5.0 (0.0)</td></tr>
<tr><td>Median 2D approach</td><td>3.0 (0.5)</td><td>3.5 (0.5)</td><td>4.0 (0.5)</td><td>4.0 (0.5)</td><td>3.0 (1.0)</td><td>3.5 (0.5)</td></tr>
</table>

The results indicate that the main positive feature for users is the learning of the interaction in both approaches, indicating that the whole mechanism is intuitive enough and no significant prior knowledge or training is required. In general terms, in the case of the 3D interaction interface, all the results are highly positive, and superior in all categories compared to the simple 2D touch approach. This fact can be related to real time visualisation and natural interaction by moving around the 3D model provided by the proposed interaction mechanism. The second section of the questionnaire shows similar results to the first one, with higher evaluation for the 3D based interaction approach. The aspect that obtained the highest score in the case of the 3D approach are the selected mechanisms to interact with the 3D interface, related directly to the possibility of observing the cube as a real object in a real environment using the interaction device as a 'window' to visualise the 3D cube.

In general terms, the 3D approach had the best evaluation in all of the questions, indicating the preference of the users for this type of interaction over traditional touch interfaces (2D approach). Also, the 3D approach presented lower median absolute deviation in almost all the questions, indicating all the users involved in the experiment agreed in their evaluation of the interface.

Regarding the qualitative analysis, the answers for sections 1 and 2 versus different users' ages are shown in Figures 3.25 and 3.26

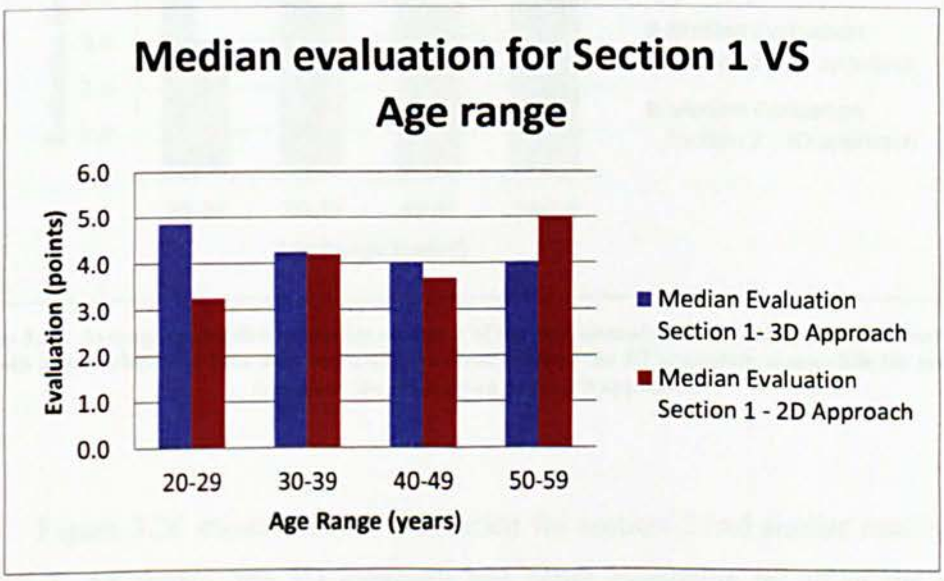


Figure 3.25: Median evaluation values for section 1 of the questionnaire (single interaction) vs. users' age for both approaches. The blue bars represent the score given to the 3D approach; meanwhile the red bars represent the score given to the 2D approach.

As shown on Figure 3.25, the users in general gave better scores to the proposed 3D approach compared to the standard touch based, in section 1 of the questionnaire, except for one case. The users of the case argued that the simple touch was more precise to aim. In general, it can be noted that users under 30 years gave the best scores to the 3D approach while the 2D touch interface was preferred mainly by users older than 50 years.

The result for the statistical significance of the data obtained in this case, $N = 10$, *Wilcoxon Statistic* = 11, $p < 0.05$ (one tailed), indicating a significant mean difference between both approaches in the results of section 1 of the questionnaire meaning there is a difference between the users preference for one method over the other, in favour of the 3D approach.

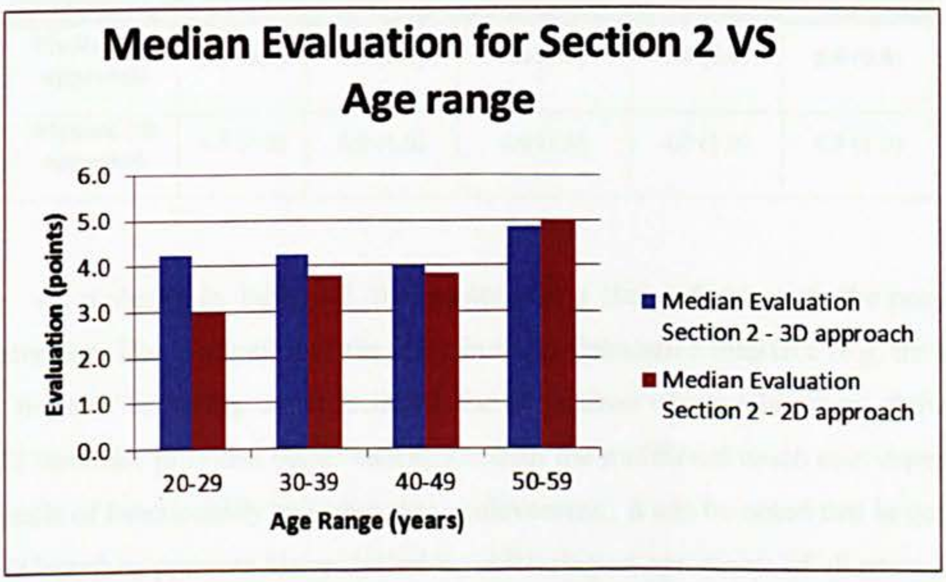


Figure 3.26: Average evaluation values for section 2 of the questionnaire (single interaction) vs. users' age for both approaches. The blue bars represent the score given to the 3D approach; meanwhile the red bars represent the score given to the 2D approach.

Figure 3.26 shows that the evaluation for section 2 had similar results with section 1. As shown, the 3D approach had better evaluation by all of the users, except with the users over 50 years, who contested the precision of the touch approach over the 3D approach.

In this case, $N = 10$, *Wilcoxon Statistic* = 2, $p < 0.05$ (one tailed), indicating a significant mean difference between both approaches in the results of section 2 of the questionnaire, in favour of the 3D approach.

According to the figures presented it can be concluded that users prefer the 3D movement based interface over the traditional 2D touch based interaction, and that is related to the natural interaction interface based on 3D movements and positioning, plus the possibility of visualising a virtual object in a real environment.

3.5.2.7.2. Collaborative interaction

The results of the qualitative evaluation of the collaborative interaction between users are presented in this section. The summarised results for the questionnaire are presented in Table 3.6.

Table 3.6: Median values and median absolute deviation in each question for both collaborative interfaces.

	QC1	*QC2*	*QC3*	*QC4*	*QC5*
Median 3D approach	**4.5 (0.5)**	**4.5 (0.5)**	**5.0 (0.0)**	**5.0 (0.0)**	**5.0 (0.0)**
Median 2D approach	4.0 (1.0)	4.0 (1.0)	4.0 (1.0)	4.0 (1.0)	4.0 (1.0)

As it shown in Table 3.7, the results give a clear advantage to the proposed 3D interface. The evaluation of the users in the collaborative interface (e.g. the users work in pairs, receiving direct feedback for the actions of the other user) show that the 3D interface provides better interaction than the traditional touch one; especially in aspects of functionality and objective achievement . It can be noted that in general, the 3D interface presents lower deviation, indicating an agreement of all users in the evaluation, which can be related with the direct 3D manipulation in a real environment.

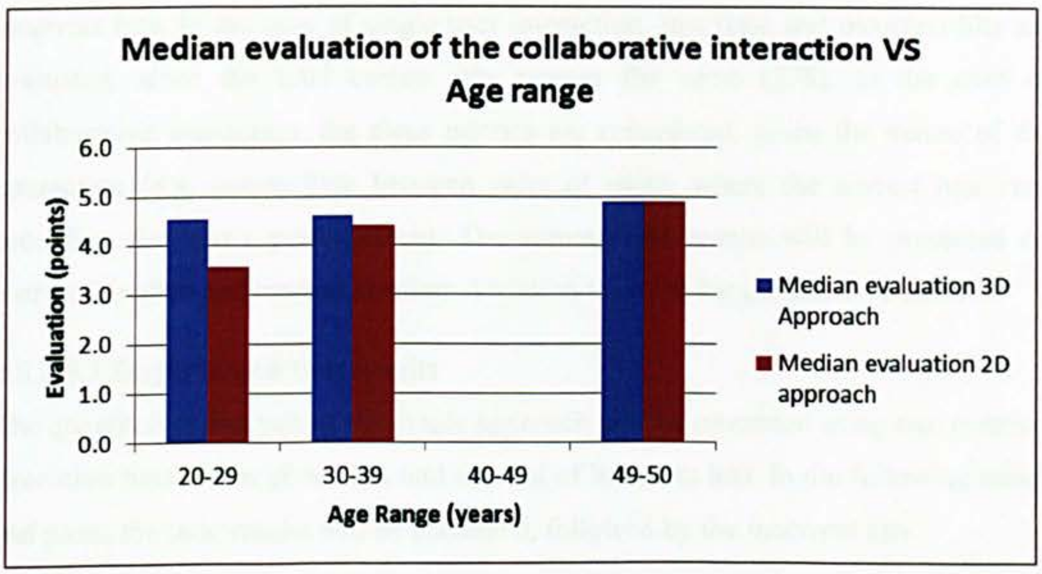


Figure 3.27: Median evaluation values for collaborative interaction questionnaire vs. users' age for both approaches. The blue bars represent the score given to the 3D approach; meanwhile the red bars represent the score given to the 2D approach.

Figure 3.27 shows the evaluation of the users versus their ages. In this experiment there were not users between 40 and 49 years old. The users in general prefer the 3D approach over the traditional 2D touch based interface. For users older than 39 the preference for both approaches is similar. The use of 3D visualisation in a real environment and the use of the virtual 3D model seem to be the key points that enhance the proposed user interface also providing a more realistic collaborative experience.

The statistical significance results for this experiment with $N = 20$, *Wilcoxon Statistic* = 2, $p < 0.05$ (one tailed), indicate a significant mean difference between both approaches in the results of the collaborative qualitative evaluation, in favour of the 3D approach.

Since the qualitative data provide the information about user preferences, it is necessary to explore the results under a quantitative point of view. In the following sections, the quantitative results will be analysed.

3.5.2.8 Quantitative results

The quantitative results presented on this section are related to three metrics used to evaluate the general performance of the users in single and collaborative interactions. The metrics used were the time to complete the task, total correct hits and total incorrect hits. In the case of single user interaction, just time and incorrect hits are evaluated, since the total correct hits remain the same (278). In the case of collaborative interaction, the three metrics are considered, given the nature of the interaction (e.g. competitive between pairs of users, where the correct hits vary according the user's performance). The summarized results will be presented on terms of median and median absolute deviation to avoid the influence of outliers.

3.5.2.8.1 Single interaction results

The quantitative analysis of the single approach will be presented using two metrics: execution time of the given task and amount of incorrect hits. In the following tables and plots, the time results will be presented, followed by the incorrect hits.

The overall times for task completion in both approaches are presented on Table 3.7.

Table 3.7: Overall median time in seconds for single user interaction (3D and 2D approach). The median absolute deviation is also shown in brackets.

	3D Approach	**2D Approach**
Median Time (seconds)	160.4 (39.3)	272.3 (62.8)

The average time to achieve the goal of the task (remove the 278 grey cubes) shows that the 3D approach is much faster than using the simple touch based interface. This can be explained by the nature of the interaction: the aiming in the 3D approach is much faster, because its more natural and requires the user to aim directly using the tablet device. Meanwhile in the case of the 2D approach, the users must move the cube using the directional touch area, which requires more interactive steps.

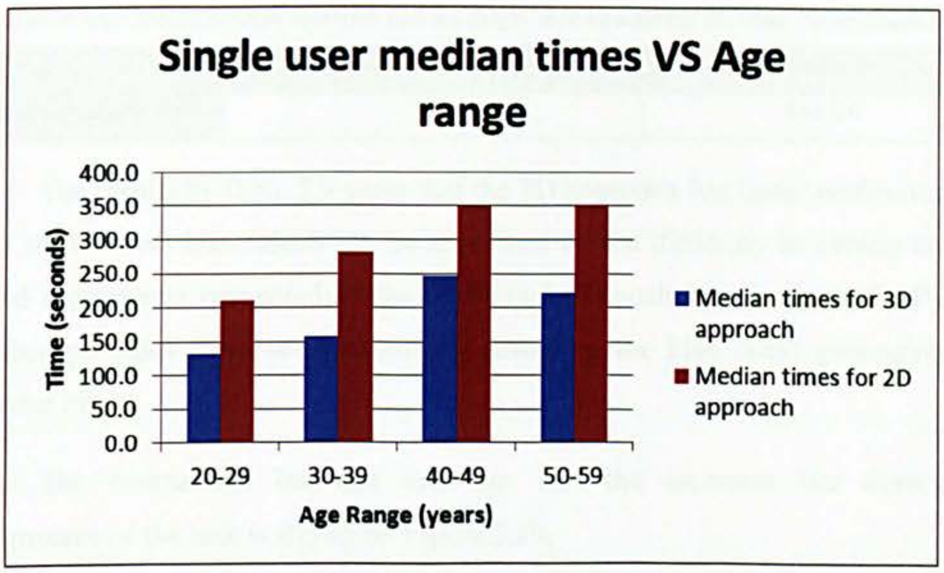


Figure 3.28: Median time to perform the task vs. users' age for both approaches. The blue bars represent the time to perform the task on the 3D approach; meanwhile the red bars represent the time to perform the task on the 2D approach.

The comparison between user age and the time they required to perform the task is shown on Figure 3.28. It can be observed that the users performed the task significantly faster using the 3D approach. This result is related to the aiming process, as discussed previously, that provides to the user a faster identification of the cube section to be removed. It can be observed that the users with the best performances are in the age range of 23 to 32 in the case of the 3D approach (a trend that is not that clear in the case of the 2D approach). Also, the users older than 40 present the worst performances for both cases but significantly faster with the proposed one.

The result for the Wilcoxon signed-rank test of statistical significance, with $N = 10$, *Wilcoxon Statistic* =0, $p < 0.05$ (one tailed), indicating a highly significant mean difference between the two approaches presented, in favour of the 3D approach.

The case of the incorrect hit's metric is analysed in the following section. The general results are shown in Table 3.8.

Table 3.8: Overall median incorrect hits for single user interaction (3D and 2D approach).

	3D Approach	2D Approach
Incorrect median hits	28.0	431.0.0

The results in Table 3.9 show that the 3D approach has better performance in terms of incorrect hits, which can be explained by the difficulty in aiming and the related movements compared to the traditional 2Dtouch based approach. The 3D interface provides more accurate aiming (avoiding the blue cues) generating less incorrect hits.

The comparison between user age and the incorrect hits during the performance of the task is shown on Figure 3.29.

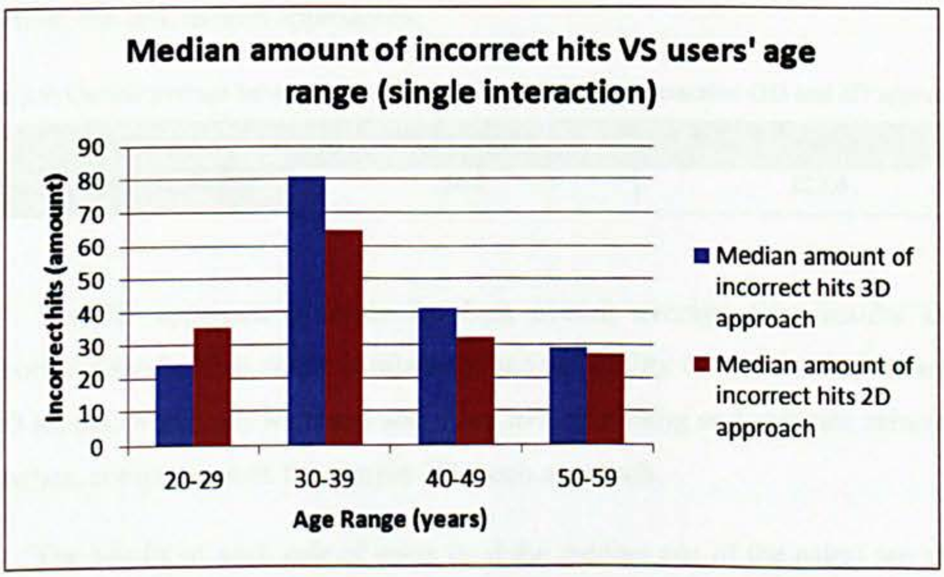


Figure 3.29: Median amount of incorrect hits during the task's performance vs. users' age for both approaches. The blue bars represent the amount of incorrect hits on the 3D approach; meanwhile the red bars represent the amount of incorrect hits on the 2D approach.

Figure 3.29 shows that fewer users have less incorrect hits in the touch based approach than in the 3D one, which can be related with the precision of aiming in. The worst performances can be seen in the 3D approach (as it was observed, because some of the users gave more importance to the correct amount of hits to "defeat" their counterparts), but also the best time results, which explain why the overall average amount of incorrect hits is less in the 3D approach.

The statistical significance in this case, with $N=10$, *Wilcoxon Statistic* = 25, $p>0.05$ (one tailed), indicates not significant statistical difference between the amount of incorrect hits between both approaches. However, given the time results of the 3D approach, the amount of incorrect hits does not affect the overall performance of the interface. The next section presents the qualitative results for the collaborative interaction in both approaches.

3.5.2.8.2. Collaborative interaction results

The metrics to be analysed in this section are the time taken to perform the task, the correct hits and the incorrect hits.

Table 3.9 presents the overall average times for all of the pairs of users that performed the task in both approaches.

Table 3.9: Overall average time in seconds for collaborative user interaction (3D and 2D approach).

	3D Approach	**2D Approach**
Median Time (seconds)	86.1	127.0

The 3D approach presents the best overall average time results in the collaborative mode. This result is related to the possibility of free movement around the 3D model, which allows faster and more natural aiming and accurate removal of grey cubes, compared with the simple 2D touch approach.

The results of each pair of users (and the median age of the pairs) are shown in Figure 3.30.

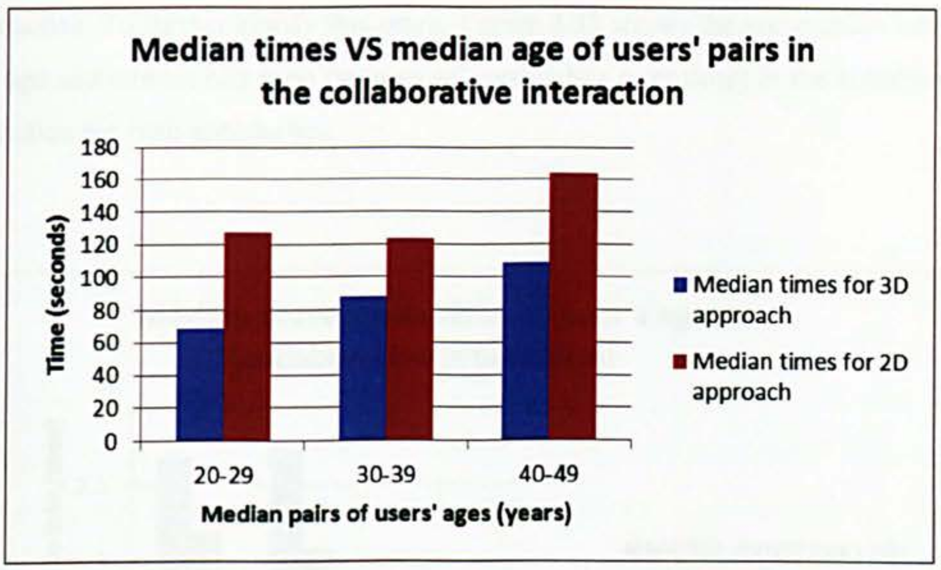


Figure 3.30: Median time to perform the task vs. pairs of users' age range for both approaches. The blue bars represent the time to perform the task on the 3D approach; meanwhile the red bars represent the time to perform the task on the 2D approach.

Figure 3.30 shows that all of the pairs of users have better results in the 3D approach than in the 2D one. This is directly related to the aiming speed, as described in the single interaction section (section 3.5.2.8.1). Also, it can be noted that in general the faster pairs of users are younger than 31 in both approaches.

The Wilcoxon ranked signed test, with, $N = 10$, *Wilcoxon Statistic* =0, $p < 0.05$ (one tailed), indicates a highly significant mean difference between the two approaches presented, in favour of the 3D approach.

The second metric to be analysed is related to correct hits. Table 3.10 shows the average results in the collaborative interaction, divided in users with highest and lowest amount of correct hits by user pairs.

Table 3.10: Overall median correct hits for collaborative user interaction (3D and 2D approach).

Correct hits	3D Approach	2D Approach
Highest scores	156.5	156.2
Lowest scores	120.5	121.5

As it can be seen, the average amount of correct hits in the case of highest and lowest amounts of correct hits shows no significant difference between both approaches. To further clarify this point, Figure 3.31 shows the comparison between user age and correct hits ratio (amount of correct hits over time) in the collaborative interaction for both approaches.

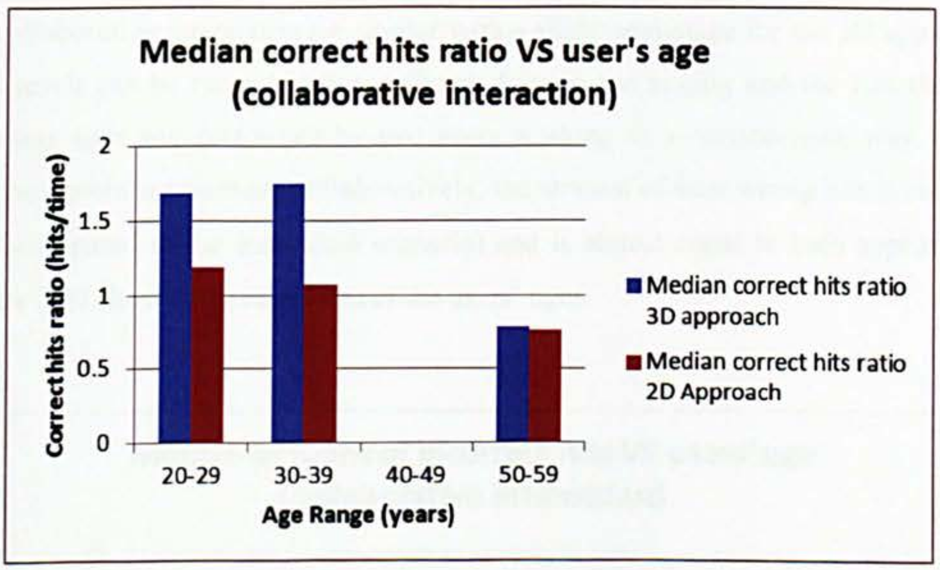


Figure 3.31: Median correct hits ratio (amount of hits per second) during task performance vs. user age range for both approaches (collaborative interaction). The blue bars represent the ratio of correct hits on the 3D approach; meanwhile the red bars represent the ratio of correct hits on the 2D approach.

As seen in Figure 3.31, the results for the 3D approach are better than the results for the 2D approach (more hits per second). This can be explained by the point argued earlier about the users' speed of aiming.

The results of the statistical significance, with, *N*=20, *Wilcoxon Statistic* = 5, $p<0.05$ (one tailed), indicate significant statistical difference between the ratio of correct hits between both approaches, favouring the 3D approach.

The last metric to be analysed for collaborative interaction is related to incorrect hits. Since the incorrect hits are not related with a total score, this metric will be analysed individually. Table 3.11 shows the average amount of incorrect hits by each approach.

Table 3.11: Overall median incorrect hits for collaborative user interaction (3D and 2D approach).

	3D Approach	2D Approach
Median incorrect hits	18.0	18.5

The results shown on Table 3.11 indicate that the average incorrect hits for the collaborative interaction are similar with a slight advantage for the 3D approach. This result can be related to two different factors: the aiming and the fact that the task was split and performed by two users working in a collaborative way. Since now two users are working collaboratively, the amount of total wrong hits is reduced (half compared to the individual scenario) and is almost equal in both approaches. Figure 3.32 shows the results versus the users' ages.

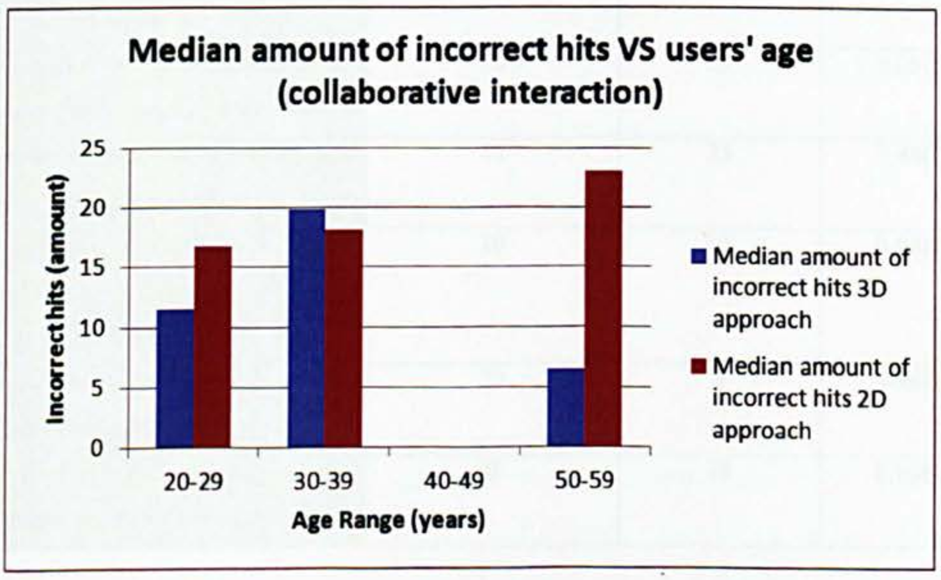


Figure 3.32: Median incorrect hits during the task's performance vs. users' age range for both approaches (collaborative interaction). The blue bars represent the amount of incorrect hits on the 3D approach; meanwhile the red bars represent the amount of incorrect hits on the 2D approach.

As shown, the simple touch approach (2D approach) presents results closer to the average value than the 3D approach that can be related to the movement speed of the users. However, it can be observed that the total amount of incorrect hits is much less than the amount shown in the single case, sustaining the idea that the collaborative approach drastically reduces the amount of general mistakes as it does with the performance's times.

The Wilcoxon ranked signed test, with, N=20, *Wilcoxon Statistic* = 78, $p>0.05$ (one tailed), indicates not significant statistical difference between the amount of incorrect hits between both approaches.

Table 3.12 shows the values obtained for our different sets of experiments.

Table 3.12: Statistical significance values for the experimental data obtained.

Experimental Data	Amount of subjects involved (N)	Wilcoxon Statistical	P-Value
Single Interaction, qualitative questionnaire Section 1	10	11	0.04648
Single Interaction, qualitative questionnaire Section 2	10	2	0.00347
Collaborative interaction – qualitative questionnaire	20	32	0.00695
Single interaction – Average time of task performance of users	10	0	0.00256
Single interaction – Average amount of incorrect hits	10	25	0.44038
Collaborative interaction - Average time of task performance of the pairs of users	10	55	0.00256
Collaborative interaction – Average amount of correct hits' ratio	20	5	0.0001
Collaborative interaction – Average amount of incorrect hits	20	78	0.16602

To close this section, a comparison of quantitative and qualitative results is shown.

3.5.2.9. Qualitative and quantitative results' comparison

The analysis of qualitative and quantitative results will be divided in two subsections: single and collaborative interaction.

3.5.2.9.1. Single interaction analysis

The results in Figures 3.25 and 3.26 show that the users in general prefer the 3D approach. The fact that the 3D approach presents lower inaccuracies in the amount of incorrect hits (see Figure 3.29) seems to affect not only the opinion of the users about the interaction, but indicates a relation with the performance times, especially in the case of the 2D approach (e.g. one curve seems to be the inverse of the other)..

In general terms it can be argued that the 3D interaction approach is preferred because provides a faster interaction, higher precision in aiming and more natural manipulation of the 3D environment.

3.5.2.9.2 Collaborative interaction analysis

The results presented in Figure 3.27 show a preference of the 3D approach over the 2D one, especially for users under 30. The trend in the time taken to accomplish the task is similar to the case of the single user interaction, where the lower performance times are related to more positive qualitative evaluations.

In the case of correct and incorrect hits, there are similar results as in the case of single interaction, but with a less clear relationship between the 3D and 2D approaches. This indicates that the amount of wrong hits does not affect user preference.

3.6. Conclusions

The description of a novel method for interaction in collaborative 3D environments using portable devices was presented in this chapter. Interaction with the system is based on 3D movement interaction in a 3D environment where 3D positioning and movement based interaction approach is provided. Initially to achieve 3D collaborative interaction, the information provided by the inertial sensors of the portable device was used. Since the information from the inertial sensors does not provide accurate enough data to place and follow the 3D virtual scenes, the system was changed and an approach based on computer vision and augmented reality was introduced to place the 3D models in a real environment (achieving enough accuracy for the proposed 3D positioning and movement-based interaction system). The evaluation and validation of this approach was performed in two stages: the first was related to the evaluation of the accuracy in 3D positioning using inertial sensors. The second stage focused on a set of experiments to compare the 3D interaction approach versus the traditional 2D touch based interfaces. The second stage was further divided in two stages: single user and collaborative interaction. The first experiments demonstrated that the inertial sensors of modern devices are not accurate enough to provide an accurate 3D interaction system, so the need of a vision based approach was evident. In our case, the vision based software was implemented using the Qualcomm's Vuforia framework that provides algorithms to perform 3D interaction in portable devices. This approach provided the necessary tools for the creation of the software capable of demonstrating and evaluating the proposed 3D interaction.

The second set of experiments aimed to retrieve qualitative and quantitative information about users' interaction with the 3D environment, covering both single and collaborative interactions. During the single interaction, the users were given the task to interact with a 3D virtual cube in a simulated sculpting game. The task is to reveal a 3D figure by removing all the cubes in the model without removing the components of the 3D figure to reveal; first using a 3D interaction approach (where the user had to move around a virtual 3D model to see all the sides of the model) and then using a simple touch approach (where the cube was able to be rotated by using a set of touch commands in a corner of the interface). The collaborative interaction was based on the same mechanisms, but this time, the users worked in pairs to perform the task.

The results of single and collaborative experiments showed that users prefer the 3D approach over the traditional 2D touch based approach, which also showed better results in the performance times and amount of incorrect hits. Also, the 3D collaborative approach showed a significant reduction in time performance and the amount of incorrect hits, which indicates the advantage of collaborative interaction interface over the single one, and also the superiority of 3D interaction over the 2D touch counterpart. The advantage of 3D interaction over traditional touch interfaces is directly related to the use of a more natural interaction approach, where the users can interact with the 3D environment as they are used to do in the real world, making the tasks more intuitive and easier to perform.

The next chapter presents an approach for database interaction oriented to overcome the problems of 3D manipulation of data for developers, removing the need of the portable devices for the interaction process. In this way, the interaction mechanisms and the overall interface are more similar to the real world interactions, aiming to demonstrate the 3D interaction advantages over traditional 2D interfaces.

Chapter 4

HCI for multi-dimensional databases

4.1 Context and Overview

Chapter 3 presented a framework to provide 3D collaborative interaction using portable devices, aiming to improve the ways humans interact with computers by generating systems that resemble real world interactions. The need for more real world-like interaction also requires mechanisms to interact in a direct way, with 3D information representations. Removing intermediate devices allows a more direct interaction between the users and the data that is desirable to improve human-computer interaction, especially when the users are developers, allowing direct interaction with data elements, improving the software metaphor understanding and development.

Due to the emergence of new hardware to interact with computers, there is also a growing need for more interactive keyboard free software. New ways to interact with machines have been suggested, based on computer vision and image understanding (Clifton et al., 2010). This effort has led to the design and development of interfaces more appropriate to operate in specific applications, where the graphical context is needed to manipulate the information more efficiently (Della Penna et al., 2013). The use of three-dimensional representations to display information such as pharmacological and molecular datasets has been widely used (Kanehisa et al., 2012; Rivera-Borroto et al., 2011; Leach et al., 2010). This is due to the simplicity of understanding the related information that they provide about biological models, which would otherwise be hard to understand under a text-based representation (Dixon et al., 2006; Hsin et al., 2011). Therefore this way to represent datasets has also started to be used in other scientific areas for information representation (Pfeiffer and Franke, 2011; Wurm et al., 2010).

Depth capturing devices have provided new mechanisms for interface systems, as was shown by Microsoft Kinect' s research progress in this area (Hernadez-Lopez et al., 2012). This technology allows the design of three dimensional interfaces that can be manipulated by the use of gestures, giving the users a more natural way to interact with information. The combination of 3D interaction with 3D representation of information also increases the understanding of the tasks related to the interface, minimising the time and amount of training sessions (Green et al., 1996).

However, the graphical representations of data on three dimensions in databases have been used poorly focusing only on storing figures (3D shapes). The new generation of interfaces requires the creation of software capable to deal with 3D representation and 3D interaction at the same time.

In this chapter a framework to interact with data elements in a 3D space is presented. The system provides two mechanisms to interact using 2D and 3D gestures based on data acquired from a Kinect sensor and on hand detection and gesture interpretation algorithms. The proposed architecture is analysed indicating that 3D interaction with information (e.g. 3D datasets) is possible, and provides advantages over a 2D interaction over the same problem. Finally, two sets of experiments were performed to evaluate 2D and 3D interaction styles based on natural interfaces compared also with traditional interaction mechanisms for 3D databases. The contributions of this work are a method for 3D database interaction, a flexible architecture layer for finger-based interaction and two handed finger-based interaction styles definition.

4.2. Previous Work

Structures to represent information in an intuitive way to understand and explore datasets have become a challenge for researchers. There is a fundamental need to create interactive tools that provide access to large amount of data, however, in order to handle modern database systems, advanced knowledge and training is essential. A new generation of databases aim to change their typical text-based representation to visual formats, where interaction can be achieved by using natural interfaces, such as gesture based commands or multi-touch interactions, instead of complex sequences of commands (Idreos and Liarou, 2013).

Technologies aimed at improving interaction with databases led to the creation of new paradigms to visualise information. The graph databases provide mechanisms that improve the classical relational model. Graph databases represent information in the shape of graphs where each node corresponds to a specific data type with specific attributes (i.e. address, date, user id, etc.), while the nodes represent connections between the data elements (i.e. source, sink). The graph databases present advantages in the retrieval of information compared to the conventional relational models, offering a new way to store and retrieve data (Vicknair et al., 2010). This type of representation also provides another advantage over relational methods. The internal representation of information is based on a graph model; the creation of interfaces capable of dealing with information under a graphic interface is intuitive and allows the use of both traditional and modern input devices, such as multi-touch and gesture based systems. The representation of information using this graph model has the problem of supporting only 2D interfaces to interact with information.

Other models of databases providing data modelling in multiple visual dimensions have been introduced. These models rely on a multi-dimensional representation of information, where the data can be perceived as a cube, where each "cell" of information contains a set of measures of interest, related with three information sources. This model is the one used by the paradigm known as On-Line Analytical Processing (OLAP), and graphical 3D interfaces, based on this method, can be created, focused on geographic and spatiotemporal data management systems (Gomez et al., 2012). Also this type of data modelling (OLAP data cube) has been successfully used to store and query real event data from sensors in smart buildings (Mehdi et al., 2013), where parameters such as temperature, humidity, luminescence and related events can be stored in a cubic cell that registers date, device and value related with the considered parameter. Even if OLAP data cubes provide a powerful tool to interact with information, their interfaces rely on 2D representation and traditional input devices, which reduce the level of effective interaction. The use of natural interface paradigms to interact with data presents an interesting alternative over traditional methods. User interfaces that support the use of gesture or touchless 3D interaction allow the better understanding and manipulation of 3D graphic contents and are applicable over different interaction scenarios, where direct touch is not possible. Web interfaces have not totally integrated yet functional touch interfaces (e.g. they still extensively use text-based interfacing and mice-clicking based interactions instead of direct touch, hovering or swiping over elements) and the use of touchless interfaces based on gestures allowing to browse content without changing the whole structure of the web content. These frameworks are able to utilise Kinect as an interaction device providing an alternative to traditional methods. Also they allow the design and development of a gesture vocabulary that can be used in "traditional" software interface, but with a separate module capable of connecting 3D gestures to complex browsing actions (Bohak and Marolt, 2013). Even when these frameworks provide an interesting approach, the connection between gesture based interaction interfaces and databases is still required.

Approaches that combine the previous methods (touchless interaction and multi-dimensional databases) appear to be the next step in data interaction development. Natural user interfaces for OLAP cube based systems are possible to be implemented using Kinect as gesture capturing device. A clear example is Data3 (Hirte et al, 2012), that introduces a new approach to interact with multidimensional databases, where the dataset itself is modelled as a 3D cube interface (following the logical data representation of OLAP structure). In this interface, the interaction with the data cube is done by using gesture detection based on body motion capturing, provided by skeleton tracking and the OpenNI framework, without using direct hand and finger based interaction. The supported gestures are basically swipes (for rotation), pushes (for selection), and combinations of them using both hands. The initial definition of gestures is done under a declarative environment (text-based programming) using the AnduIN data stream engine to process the events coming from Kinect translating them in command gestures. This approach makes use of a 3D interaction and 3D data representation, which allows better understanding and faster user task performance. The main issue with this approach lies in the use of full body motion to generate the gestures that actually can be a problem in desk-based applications, where a direct hand gesture based interface would be more appropriate, allowing the users to perform the same tasks in 3D in a more efficient way and with less effort.

In the following section, a finger based two handed gesture interaction method is presented not connected with the full body detection, in order to overcome the issues of the methods previously presented.

4.3. Proposed Methodology

The proposed methodology defines a common development framework for two hands-gesture interaction in 3D environments. The application is divided into layers, each of them containing specific tasks. The layer architecture and the connection between their components can be seen in Figure 4.1.

The architecture presented has similarities with a multi-touch architecture, such as the one presented by Echtler and Klinker (Echtler and Klinker, 2008) but with several important improvements and modifications. The Hardware Data Acquisition layer takes the information directly from the device previously identified by the associated API, which in our case is Microsoft Kinect (Kühn, 2011). In multi-touch architectures, this task is performed by two different layers, following the touch user input/output (TUIO) protocol (Kaltenbrunner et al., 2005). Also, depth detection is crucial during the performance of 3D activities, necessary for natural 3D interactions in indoor environments (Smisek et al., 2013). The Hand Gesture Acquisition layer is responsible for defining and identifying hand's shape, fingers and interpretations, which are related physically (fingers and hand relative relation). This process is not performed in multi-touch architectures, because all the interactions over the surface correspond to fingers and the correlation with the hand is not necessary. Furthermore, in this layer, the system selects the features that define the palm and then the ones that represent the fingers. The Gesture Interpretation layer works in a similar way to a multi-touch's interpretation layer, but in our approach, the hand position is also used to define the gestures, "translating" a gesture to a specific command, according to the interacting environment and the fingers' identification in the previous layer. The Command Graphic Association Layer makes visible the action, the association between the "logic" object and their graphic representation has to be performed in this layer. The information is passed immediately to the Graphic Interface layer, which takes control of the actions and changes in the environment after the performance of a predefined gesture, triggering a subsequent action. Finally, the graphic interface displays the outcome of the interaction. This layered architecture provides flexibility to define several combinations for 3D interactions for different applications. Also, it can provide a high degree of hardware independence, since the modular definition of the architecture allows replacing components in any layer.

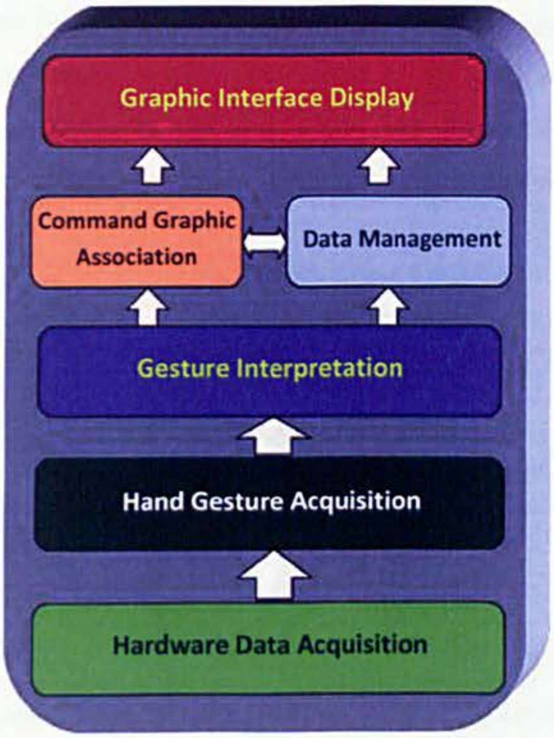


Figure 4.1: Layer architecture for two-handed gesture based systems.

4.3.1. Finger tracking

Finger tracking is related to a hand-gesture based interaction, since the proposed gesture is based on hand and finger correlation; and hands relative position. The detection of the hand is based on the depth map provided by Kinect, using an approach similar to the one presented by Xia (Xia et al., 2011), focusing on the extraction of the palm of each hand and the detection of the hands' contour using the segmented depth map. In order to identify and label each hand, the relative position in the detection space is considered. The hand in the right portion of the detection space will be the right and equivalently for the left hand. The positions are defined this way to provide the user direct feedback over the actions and the areas for each hand are defined from the centre of the detection space. If the hands cross over, since the hand detection is not connected with a skeleton tracking, the hand identification is switched, making it wrong. The system was implemented using an approach similar to the one presented by Frati (Fratti and Prattichizzo, 2011) to perform the detection of the fingers, where the hand needs to be held facing the device. The detected features can be seen in Figure 4.2 and the algorithm provides the fingertips of the index and the thumb, which correspond to the start and end points of the convexity (that corresponds to the area where the shape of the hand has gaps, allowing the separation between fingers, identifying them individually).

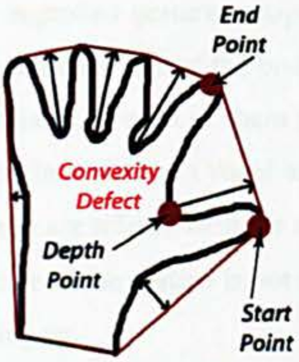


Figure 4.2: Fingers' detection algorithm using convexity, used by Fratti (Fratti and Prattichizzo, 2011).

Each finger is recognised by the detection of end points and overall convexity. Each end point represents a finger, and the palm point will provide information about the relative 3D position of each finger on the hand, allowing the performance of 3D gestures improving the interaction. However, since the detection of the hand is provided by the segmented depth map, the problem of hand gesture detection is highly dependent on the distance from the sensor, as was also discussed by Tang in (Tang, 2011), mainly due to the occlusions and possible reflection problems. The interaction space for our approach is between 0.6m and 0.9m from the camera and the width of the interaction space is about 0.6m (the hands should face towards the camera in that space). The detection of both hands can be seen in Figure

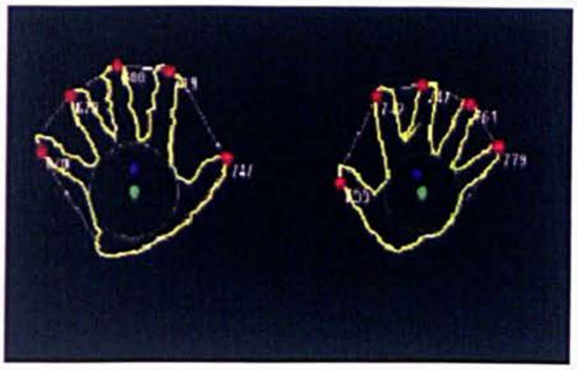


Figure 4.3: Detection of both hands under our approach.

It's necessary to mention that hands are detected as independent elements, not connected to the whole body, which further limits the detection range, but provides more degrees of freedom and improved gesture recognition speed, since there is no need to calculate the rest of the articulations of the body. As a result, the detection of the hands and the fingers associated to each of them provides enough functionality for 3D hand-gesture interaction in real time (Vogel and Balakrishnan, 2005). The main advantages of this approach are related with the absence of a training period of the system. Also, the performance of the system is not affected by hand size changes, given the model to detect the fingers.

Regarding the performance, compared to the entire human body, the hand is a smaller object with more complex articulations and more easily affected by segmentation errors. Nevertheless novel algorithms (Ren et al., 2011) match the finger parts and not the whole hand, distinguishing better, hand gestures. The accuracy of state of the art methods (Ren et al., 2013; Li, 2012) for hand and finger gesture recognition is more than 93% with an average latency of 0.075 seconds per frame, which makes this technology suitable enough for real-life HCI applications. Of course new hardware devices could improve significantly the overall performance and consequently the quality of the interaction.

4.3.2. The interaction

The interaction in our approach, during the experiments is based on hand-gestures, and as a result the interface operates using only the hands, using a depth and video capturing device (in our case, Microsoft Kinect). During the gestures' definition, two sets were developed: a set of gestures that combine both 2D and 3D movements and a second that utilises only 2D gestures, which were performed during the experiments. The choice of the two sets of gestures was set to test whether the 3D gestures can improve user interaction times and performance compared with 2D gestures (which are commonly seen on traditional touch surfaces). The proposed hand-gesture interactions are based on the number and the position of the fingers. Changes in their position trigger different actions and responses at the system. For both sets of gestures, one of the hands indicates the function (or mode) and the other performs the action, which allowed avoiding gestures' mistakes according to previous experiments performed. Considering that, the interactions are divided into three types:

- Movements: These actions correspond to changes in the position and/or the orientation of 3D graphic elements in 2D or 3D space. The action is performed only if an object is already selected and is related to the actual location of the hand in the 3D space.
- Selections: The selections are applicable only to specific 3D elements in the environment and when they are successfully performed; some components or parts of them are highlighted. The selection process is based on two actions: locate the element that will be selected and the selection process itself.
- Executions: Interaction related to performing a particular action not defined as a previous one. These actions could be the result of a combination of the previous ones or just a single hand gesture.

For our experiments, these interactions will be further analysed regarding their implementation in the results section. Also, it should be mentioned that these interactions are enough to perform the required tasks in a database system, since the actions that a standard 2D mouse can perform are a subset of them. About the insertion mechanisms, other modalities may be utilised such as speech recognition or predefined entities represented with 3D graphical elements. Also, the scrolling action can be performed in the same way as in the case of a mouse by combining the selection and moving interactions allowing the users to observe more data entries and information in general. Consequently, in the case of 3D databases the proposed interaction mechanisms are enough, supporting similar actions with a standard mouse but in a three dimensional space.

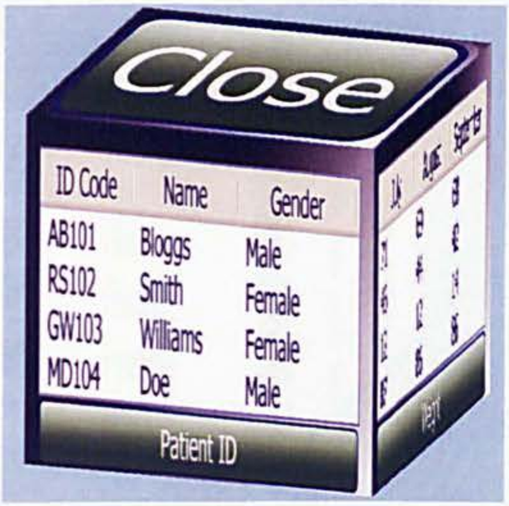
4.3.3. Three Dimensional Databases

3D databases are a derivation of multidimensional databases and in our case, the type of a cubic database is considered. These kind of databases provides an interesting field of development and research due to the data mining features offered by this model (i.e. find correlations between data elements invisible in a 2D relational model, such as the relation between items sold, stores and dates over a multi-store company database) (Jaday and Panchal, 2012). Cubic databases are useful in cases where relationships between different pieces of data are not totally clear and the connection of several information sources is required, which cannot be performed easily by the traditional 2D databases. An example is related to medical information, and the need to find associations between not obviously related features improving the diagnosis process and the patient's healthcare. As a result, important parameters for the diagnosis of diseases can be estimated, allowing a more efficient control of the demanding health services, especially for primary care (Ludwick and Doucette, 2009). For example a 3D dataset could store personal information of patients in the first table/dimension. A set of measurements with related information for each one could be available in the second table/dimension, and finally the actual measurements over a certain period of time could be part of the third table/dimension. However, due to the complexity of the traditional interaction models, we defined a novel approach that resembles the functionality, where the cube is formed by multiple tables linked together.

4.3.4. Suggested interface model

The main interface used to analyse our proposed methodology was a 3D data interaction model. In more detail, a simplified model of a cube database with multiple faces was introduced, representing information about a group of patients. The general interface model used for the evaluation of the proposed interactions can be seen in Figure 4.4. The interface can also manipulate larger and more complex databases by adding sliders that could be manipulated by a combination of a selection and moving gestures.

In order to provide a more intuitive interface, only two successive sides of the cube are displayed at any time instance (the front face with the patients' details and the right side with their measurements e.g. weight per month).



ID Code	Name	Gender
AB101	Bloggs	Male
RS102	Smith	Female
GW103	Williams	Female
MD104	Doe	Male

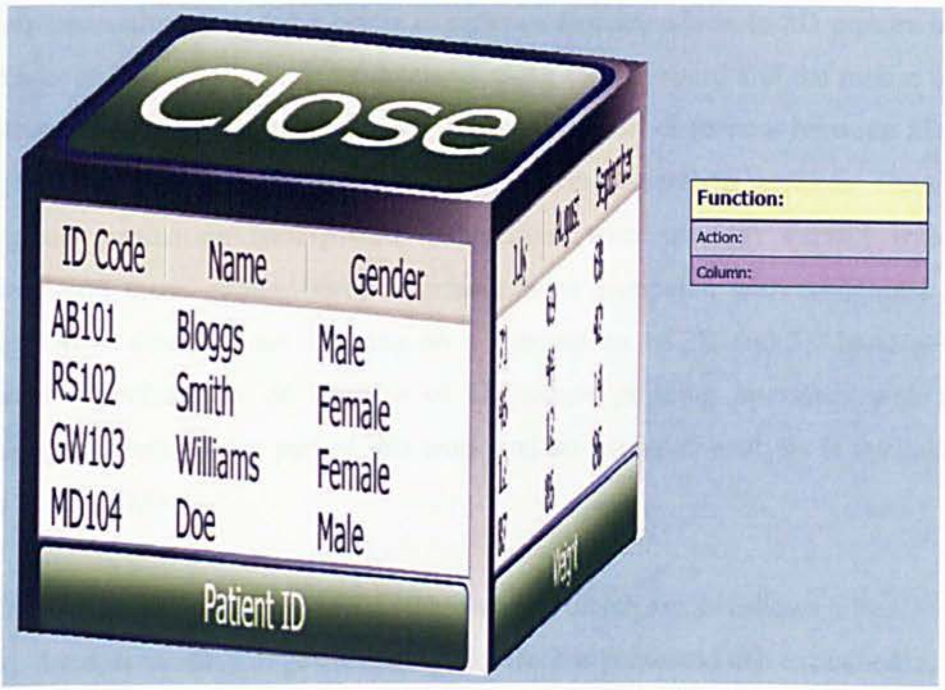
Figure 4.4: Data Cube model, one side contains personal details and the other relevant values over time.

In more detail, one face of the cube contains basic personal information of the patients. The other faces have information about the weight or other measurements of the patients, for a period of several months, (e.g. July, August and September). The top face of the cube provides the option to terminate the application by performing the related gesture over the 'close' button.

The user is able to interact with this cube using both hands. The left hand is the function 'indicator', while the right hand actually performs the action on the screen. This configuration was selected in order to limit possible confusions between functionalities and also it can be reversed to facilitate both left and right hand users. The hand of the user and particularly the index finger is followed by a screen indicator (e.g. highlighted cell) to allow the users to have a visual representation of their exact position on the screen and on the cube.

To improve the feedback to the user, a visual text chart has been added to indicate the current function performed, which changes according to the detection of the hand-finger gestures. This text indicator will show the function mode (e.g.

movement or selection according the indicator hand), if the action is being performed (selecting or moving) and finally, in which column the action is being performed. Figure 4.5 presents the cube database interface previously described.



ID Code	Name	Gender
AB101	Bloggs	Male
RS102	Smith	Female
GW103	Williams	Female
MD104	Doe	Male

Figure 4.5: Data cube model for experiment and function indicator mini screen.

In the following section the experiments related to the cube database model are presented, the set of gestures and the experimental procedure are described focusing on the users performing basic information tasks with their hands.

4.4. Experiments

In our experiments, two set of gestures were designed and developed mainly for interaction with 3D databases. The suggested sets of gestures were tested in order to demonstrate and evaluate the users experience and indicate the need for 3D interaction in these applications. The evaluation process was focused on an example of a simplified version of a 3D cube database containing information about patients and their measurements over a period of time. This simplified model considers several patients providing personal details (ID Code, Name and Gender); a list of measurement features (e.g. weight, height, heart rate, etc.) including their importance (e.g. weight) and finally, the actual values for each feature of each patient over a period of few months (e.g. July, August and September).

With these experiments we wanted to investigate whether the 3D hand gesture interaction provided a better experience in comparison to 2D gesture based interfaces and also traditional mechanisms using the keyboard and the mouse based on Structured Query Language (SQL), where the main difference between 3D and 2D gestures lies in the use of depth of the fingers to perform gestures. Therefore, during our evaluation hand-gesture interactions were used to interact with the database and these gesture-based interfaces were compared with traditional SQL queries. Also, since we are focusing on a comparison of 2D and 3D hand gesture interaction mechanisms, an analysis of 2D mouse pointing interfaces with hand movement systems is not part of this work and an extended analysis is available in (Pino et al., 2013).

The experiments are divided into three stages, which are as follows:

- the ***presentation stage*** (where the interface is presented and explained)
- the ***practice stage*** (where the users can interact with the interface and use the available features), and
- the ***task execution stage*** (where the users perform the task and quantitative factors are recorded, such as task execution time, the number of users and system errors). The defined task for the evaluation was the columns' selection from the cube database simulating a querying procedure.

In order to provide better feedback to the users, another graphic element was added to the interface: a dialog box that indicates the columns correctly selected (according to the given task) by the users (on the bottom of the screen, as it can be seen in Figure 4.10). The main elements (according to the stages' definition) of our experiments are analysed in the following section.

4.4.1. Set of experimental Gestures

Two sets of gestures were used during the experiments, as mentioned in the previous section. The fingers combination was selected after several tests with the graphic interface.

The set of the selected gestures are enough to perform all the interactions analysed in section 4.3.2 and are further divided in 3D and 2D gestures, which correspond to the first and second interaction experiments, respectively. The amount of hand-finger combinations used in each gesture was determined experimentally to avoid the confusion between gestures, allowing the correct identification of the performed action, following further experimental results about finger detection reliability (Ren et al., 2011; Li, 2012; Lee and Tanaka, 2013). Also it was observed from the initial experiments that using numbers pointed by the fingers was more intuitive at this stage helping to memorize the available interaction mechanisms. For the 3D set of gestures, the related actions are defined below:

- **Rotation:** The cube can be rotated from left to right and vice versa around the vertical axis. The rotation action is performed by keeping the left hand totally open (all the five fingers, indicating the "Rotation Mode") and simultaneously moving one finger (any, but the index finger is preferred) of the right hand from left to right or vice versa, depending on the face of the cube that the user wants to see. During this action the cube rotates smoothly from one side to the other according to the finger's position. There are no rotations about the horizontal or depth axis. Other combinations of movements to provide rotation around other axis was considered, but to simplify the interaction process and to not confuse the user, the rotation feature was limited to rotations about the Y axis.
- **Selection:** The selection is considered more as a mode than an action allowing the identification of graphic elements to be selected. In order to enter in this mode, the users must show two fingers of their left hand and place the cursor over the selected element, using the indicator finger.
- **Clicking:** The clicking action works as the execution phase of the selection mode of an identified element, and because of that during the clicking process, the user must remain in selection mode (two fingers of the left hand have to be visible). The clicking is performed by placing the indicator finger

over a selectable element and "pushing" (moving forward, towards the screen). The clickable elements on the cube are the "close" button, the column headers and the data rows. In order to choose a full column (to perform a specific task), it is just necessary to click on the column header.

The set of 2D gestures are defined below:

- **Rotation:** The rotation gestures are the same as the one for the 3D approach, because it does not have any 3D interaction itself. The combination of fingers and sequence of movements are the same for this experiment.
- **Swiping:** To perform a selection in the 2D cube interface it is necessary to swipe over the column or row. This process requires first the indicator to be positioned on the top or bottom of the column and then swipe along the column to be selected successfully. To avoid a wrong selection, the column is divided into several equivalent surface sections that must be swiped sequentially using a vertical (or horizontal) movement to perform a column (or row) selection that can be also performed from bottom to top or vice versa.

For each of the above actions, a set of thresholds is defined to indicate the initial and the final positions that will determine their range. These thresholds were selected experimentally and are dynamically based on the relative position of the person and the acquisition system. Figures 4.6 and 4.7 help to explain how the thresholds were defined.

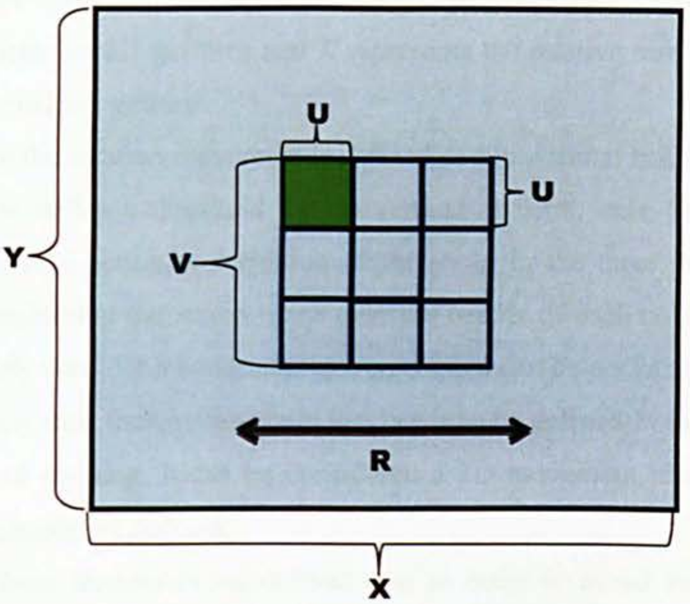


Figure 4.6: Interaction areas for the experimental interface, where the external light blue square represents the whole interaction area (with measures $X \times Y$), the cube interaction area (where $V = 3U$, with a total area of $V \times V$), the column interaction area ($U \times V$) and the individual cell interaction area ($U \times U$), where R represents the range of movement for rotation, that can be performed anywhere in the interaction area..

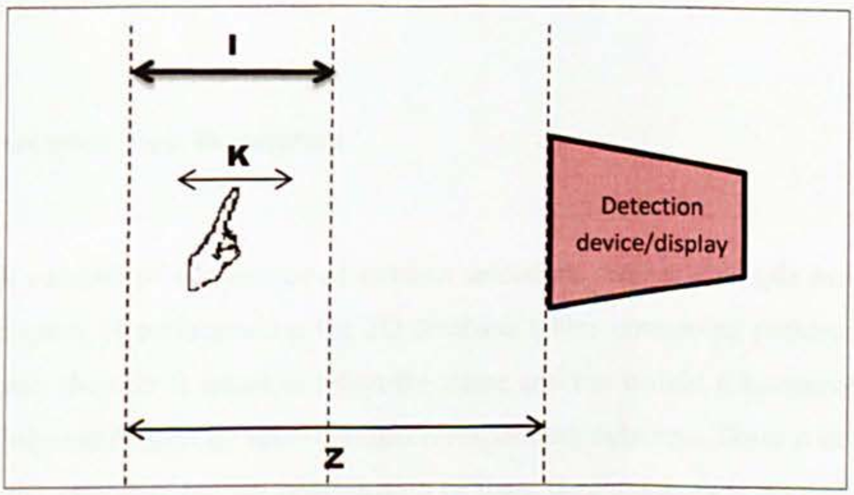


Figure 4.7: Representation of the interaction area for the interactions in the depth axis, where Z represents the maximum distance from the detection device to the user, I represents the interaction area for 3D gestures of the user and K represents the range for the clicking gesture.

Figure 4.6 shows a representation of the screen interaction area (graphic interface) with a height of Y and a width of X (in our case X=Y), where the current cube interaction area has height equal to V and width V (with $V = 3U$). The defined cell interaction area is a square of length U. R represents the range of movement to perform the rotation gesture. Figure 4.7 presents the depth interaction area, where Z

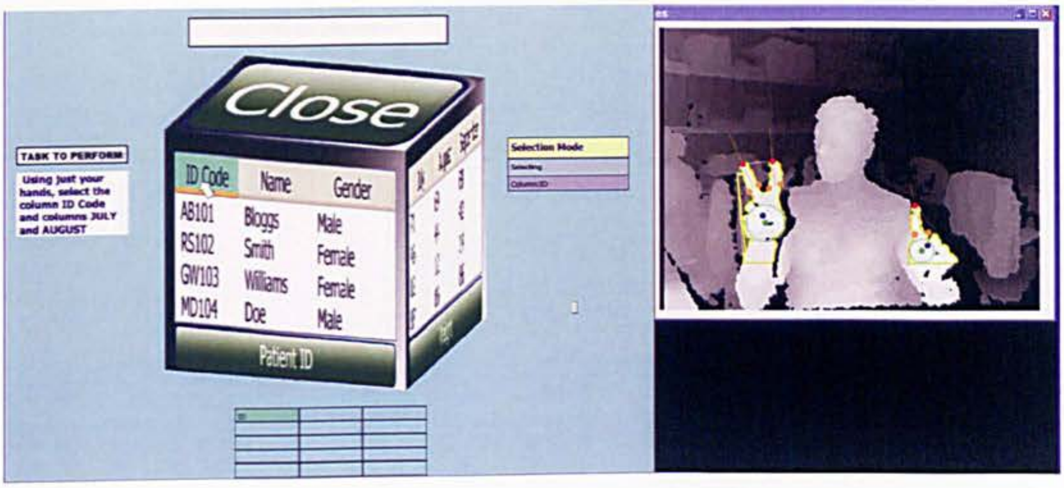
represents the maximum distance of the user to the detection device, I represents the interaction area for 3D gestures and K represents the relative size of the interaction area for the clicking gesture.

Since the rotation movement is defined as a horizontal hand movement, there is no need to define a threshold for the vertical or depth axis. Since clicking is a three-dimensional action, a definition of threshold in the three interaction axes is required, considering the action range over the header of each column. The rotation gestures solely consider a horizontal movement that can be performed at any place in the interaction area, thus its threshold just needs to be defined in the horizontal axis. In the case of swiping, it can be considered a 2D movement, thus horizontal and vertical thresholds are defined.

All these thresholds are defined also in order to avoid interference due to random movements or actions. Furthermore, the full interface has visual indicators for each task, such as the "function mode" indicator, which will show the current function, the action that is performed and the corresponding selection, as seen in Figure 4.5.

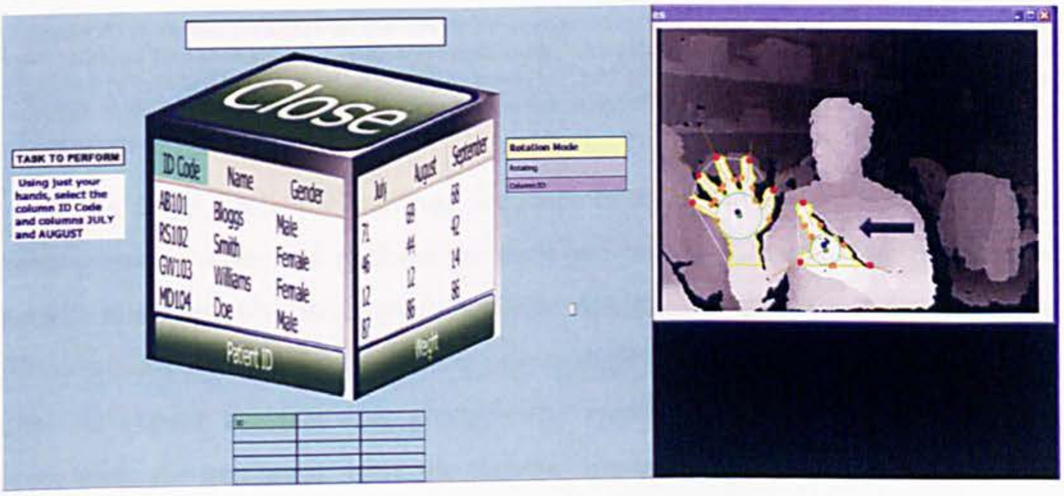
4.4.2. Execution Task Description

This task consists of a sequence of column selections, where a simple information selection query is performed on the 3D database tables containing patients' details. In this case, the user is asked to select the name and the weight information for the months July and August by selecting the corresponding columns. There is no specific order in the selection, but the combination of these data columns is needed for each face of the cube to complete the task. In the case of selecting the wrong data, the user is requested to repeat the task. This aspect reduces the possibility of a random selection allowing the user to focus on the required tasks. The ideal interaction sequence to perform the task for our 3D approach can be seen in Figure 4.8, to clarify how the interface works.


TASK TO PERFORM

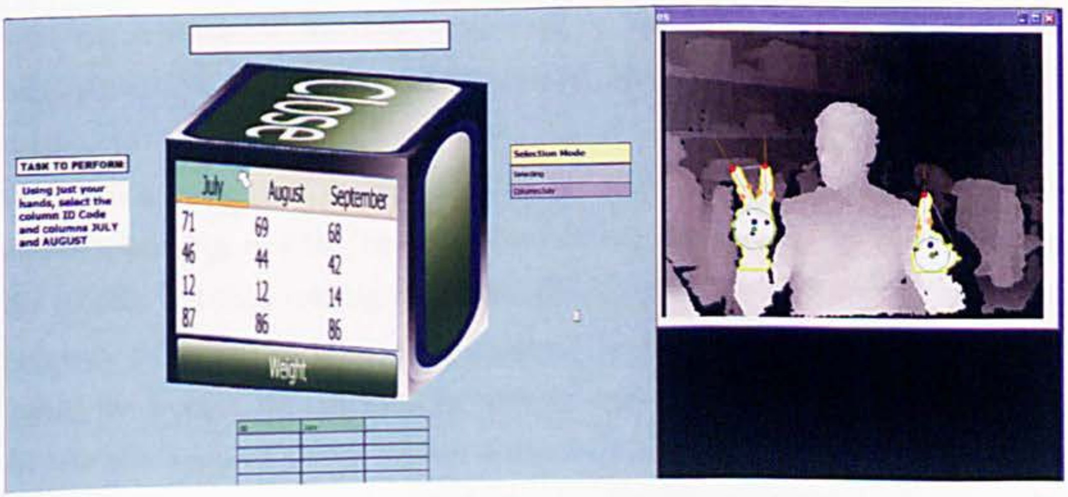
Using just your hands, select the column ID Code and columns JULY and AUGUST

(a)


TASK TO PERFORM

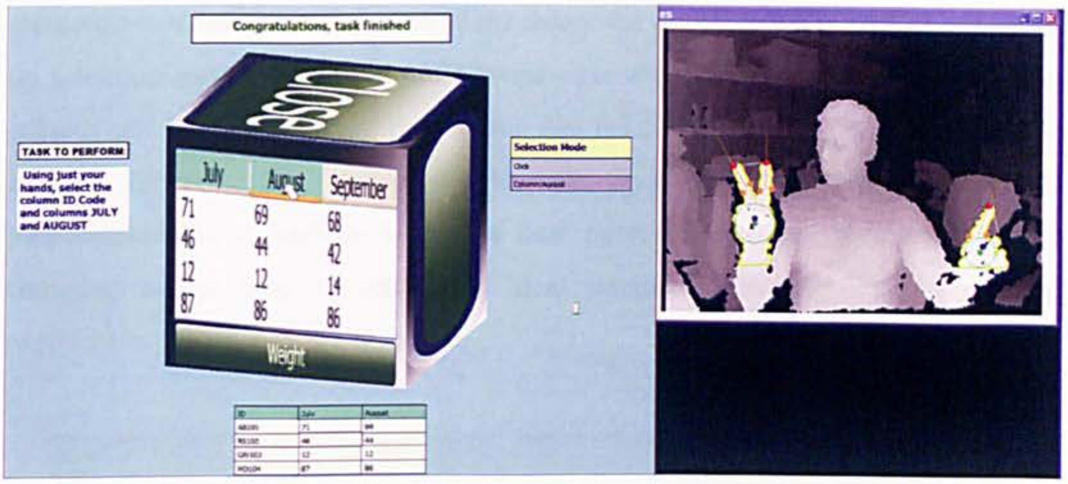
Using just your hands, select the column ID Code and columns JULY and AUGUST

(b)


TASK TO PERFORM

Using just your hands, select the column ID Code and columns JULY and AUGUST

(c)


Congratulations, task finished

TASK TO PERFORM

Using just your hands, select the column ID Code and columns JULY and AUGUST

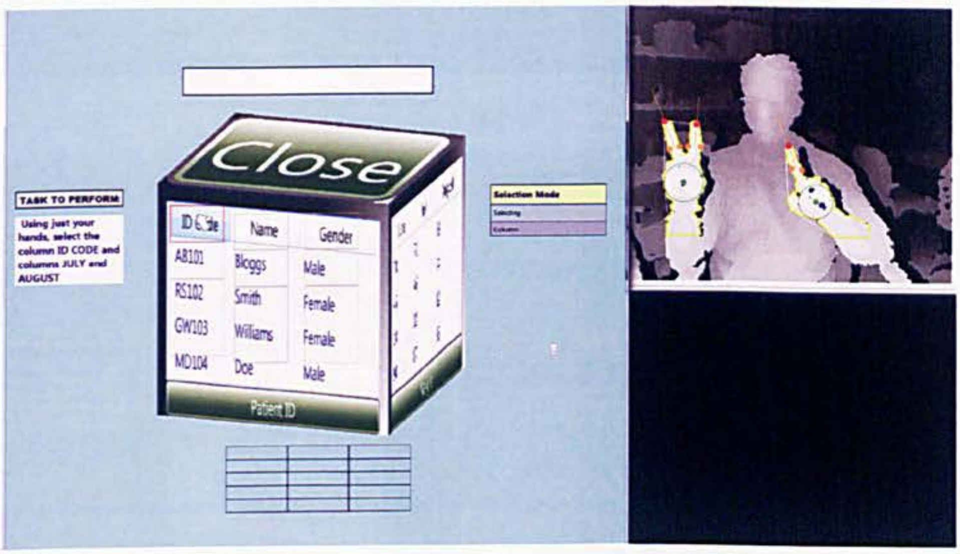
July	August	September
71	69	68
46	44	42
12	12	14
87	86	86

(d)

Figure 4.8 Interaction sequence for the task in 3D gestures case (ideal scenario): a) First step: in selection mode, click on ID code b) Second step: in rotation mode, rotate the cube (moving the indicator finger from right to left, inverted in the captured image) to access the next table of the cube (months) c) Third step: in selection mode, click on July d) Fourth step: in selection mode, click on August and the task is completed.

In more detail, the expected sequence of steps for this experiment is: first enter selection mode and perform the click over the ID Code column (there is no need to rotate the cube before performing the task, because the table that contains the ID code is shown by default when the system starts), then it is necessary to rotate the cube to expose the face that contains the months' table. Once the rotation is completed, the user must select the months July and August. During the selection process, the user has constant feedback about the performed actions and when the columns selection is correctly completed; in the lower part of the screen a highlighted table with the name of the selected column is shown. Also, the header of the selected column changes colour in the case of a correct clicking. Once the task is successfully completed, the selected data are displayed on the lower part of the interface (see Figure 4.10). However, the user can start rotating the cube, selecting the months and then rotating back and selecting ID Code or following any other sequence of actions to achieve the expected. If the user selects a wrong column during the process, the task must be restarted again. This rule was placed to ensure that the user's actions are not random and to achieve a fair comparison.

The 2D interface works in a similar way but the main difference is that all the interaction is performed like on a touch device, which means there are no depth related movements. The selection instead is made by swiping over the data (e.g. columns) as was described previously without any 3D interactions. This swiping must start in a defined initial place of the data to be selected (in the case of a column, on selection mode, the finger must swipe over the top or bottom sections of the column and then move along the column and reach the opposite extreme). Also, to improve the feedback provided to the user, another feature was added: semi-transparent cells to indicate when the user passes over them were implemented changing colour (e.g. to red). The ideal interaction process for this second experiment can be seen in Figure 4.9.

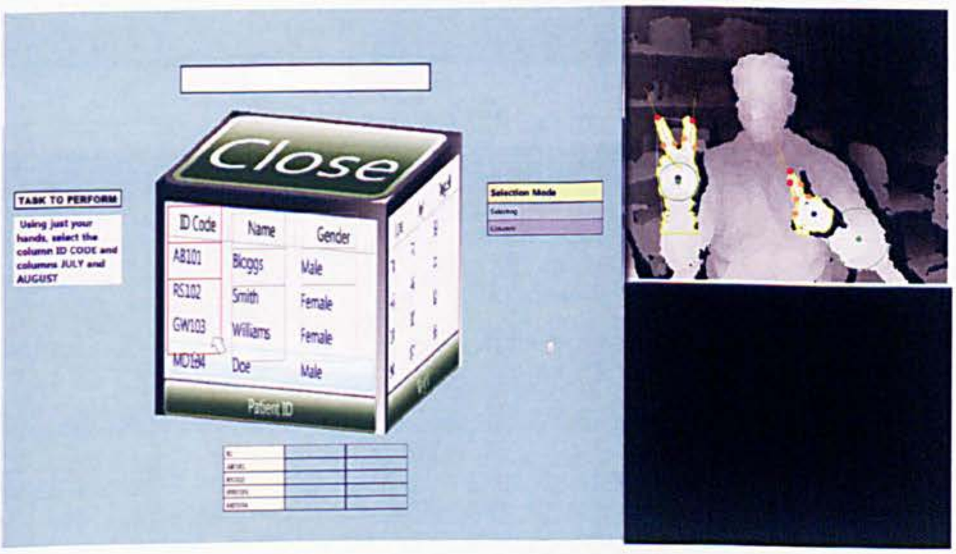


TASK TO PERFORM

Using just your hands, select the column ID CODE and columns JULY and AUGUST

ID Code	Name	Gender
AB101	Bloggs	Male
RS102	Smith	Female
GW103	Williams	Female
MD104	Doe	Male

(a)

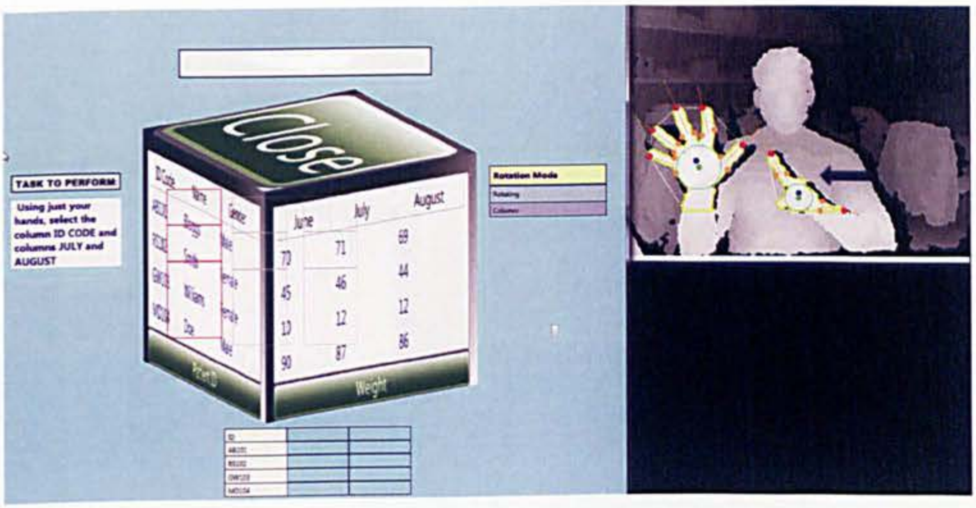


TASK TO PERFORM

Using just your hands, select the column ID CODE and columns JULY and AUGUST

ID Code	Name	Gender
AB101	Bloggs	Male
RS102	Smith	Female
GW103	Williams	Female
MD104	Doe	Male

(b)



TASK TO PERFORM

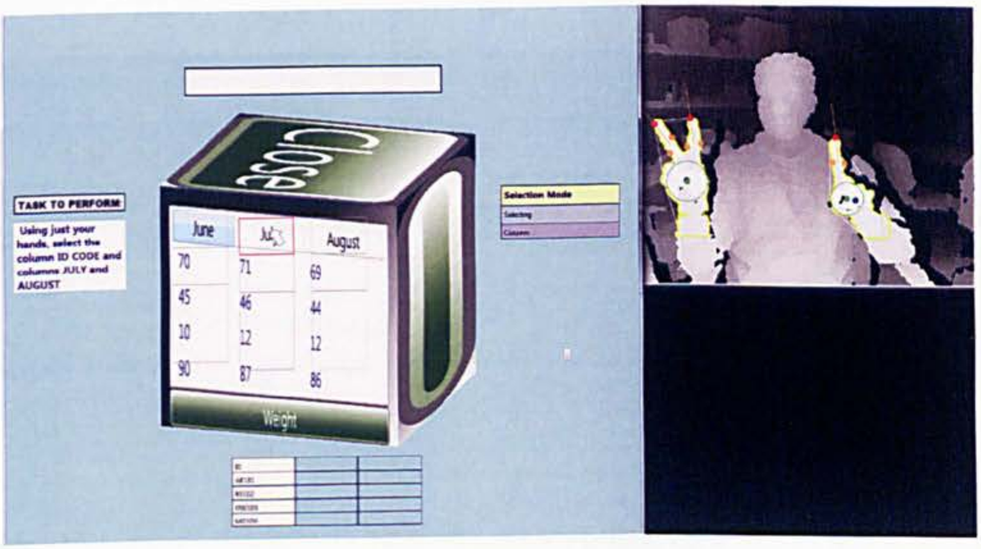
Using just your hands, select the column ID CODE and columns JULY and AUGUST

Rotation Mode

Rotating

Columns

(c)



TASK TO PERFORM

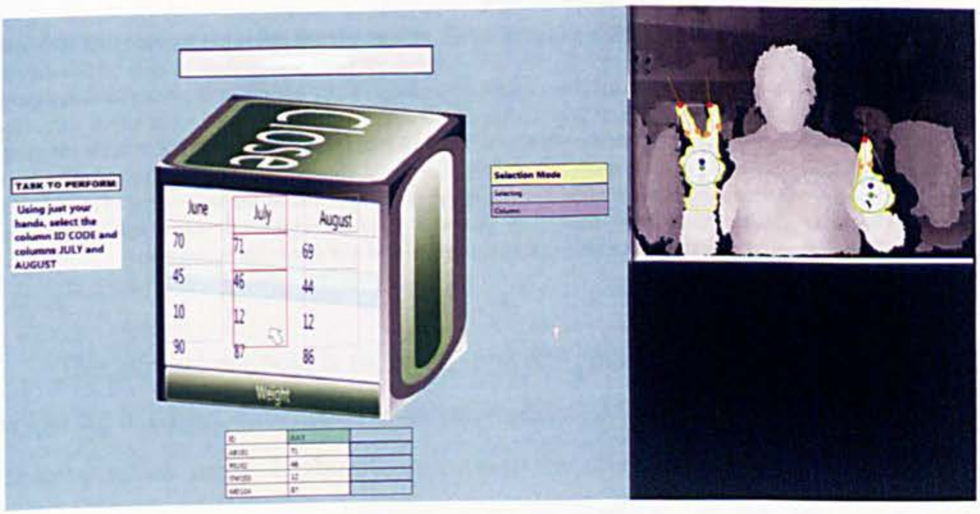
Using just your hands, select the column ID CODE and columns JULY and AUGUST

Selection Mode

Selecting

Columns

(d)



TASK TO PERFORM

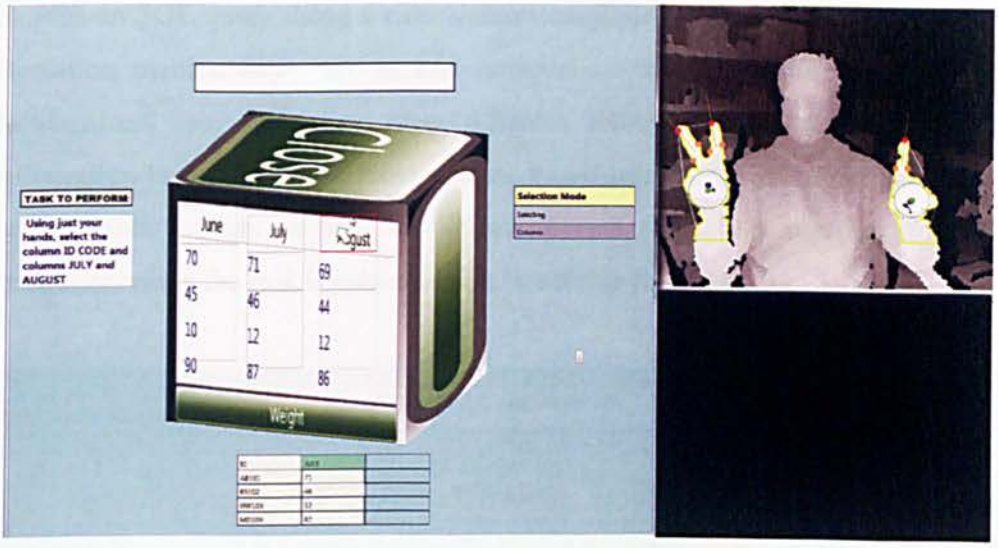
Using just your hands, select the column ID CODE and columns JULY and AUGUST

Selection Mode

Selecting

Columns

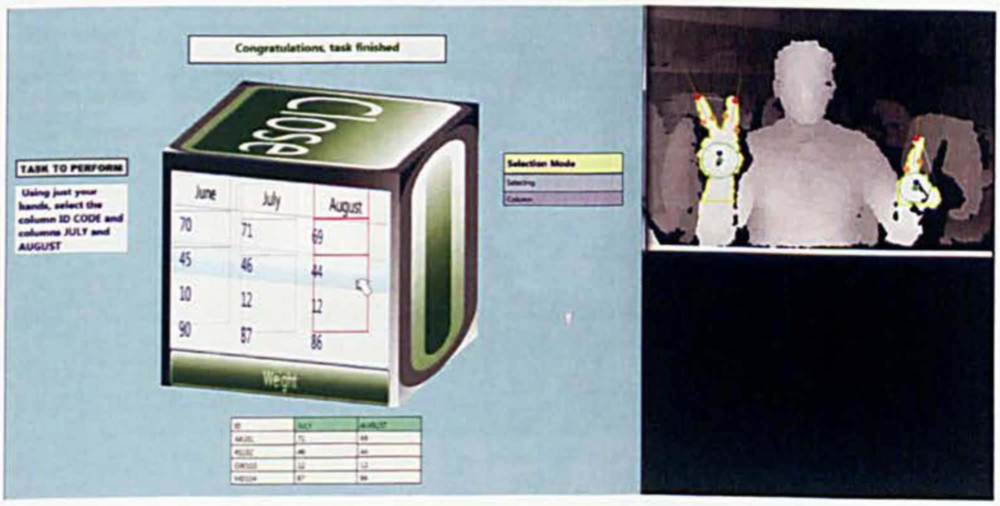
(e)


TASK TO PERFORM

Using just your hands, select the column ID CODE and columns JULY and AUGUST

June	July	August
70	71	69
45	46	44
10	12	12
90	87	86

(f)


Congratulations, task finished

TASK TO PERFORM

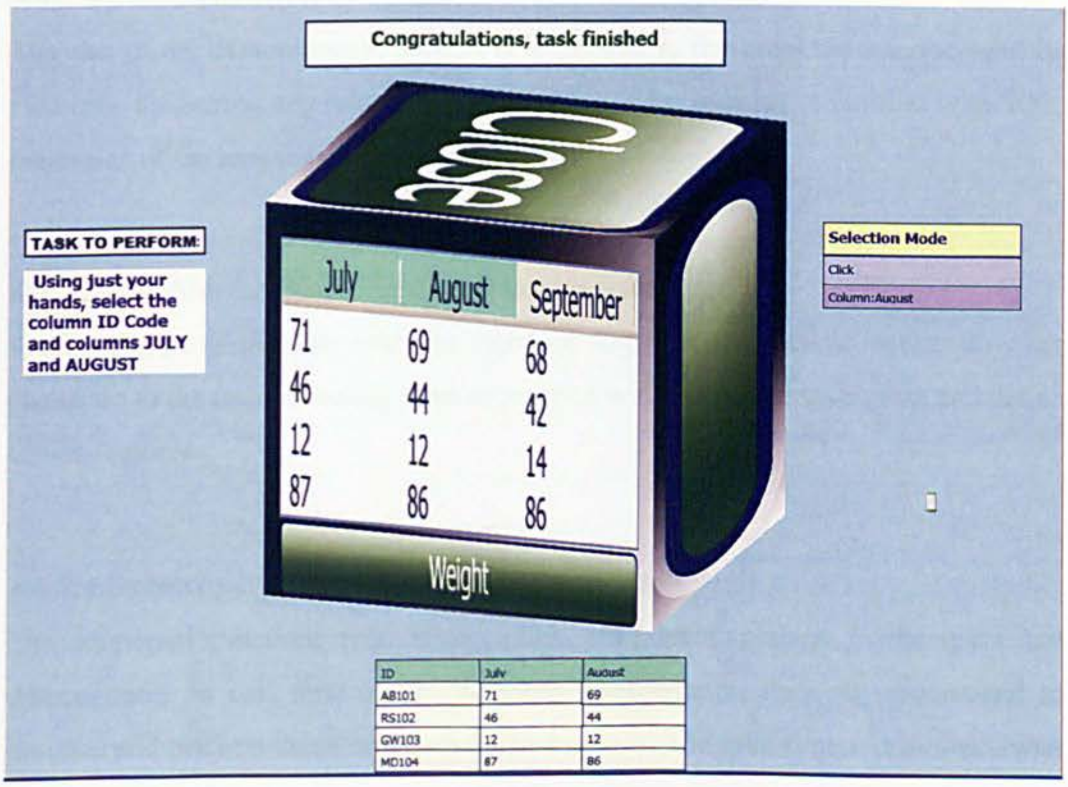
Using just your hands, select the column ID CODE and columns JULY and AUGUST

June	July	August
70	71	69
45	46	44
10	12	12
90	87	86

(g)

Figure 4.9: Interaction sequence for the task in 2D gestures case (ideal scenario): a) First step: in selection mode, place the cursor (using the indicator finger) over the ID Code column's header b) Second step: keep swiping vertically over the column until reaching the lower section of the column c) Third step: in rotation mode, rotate the cube (moving the indicator finger from right to left, inverted in the captured image) to access the next table of the cube (months) d) Fourth step: in selection mode, place the cursor (using the indicator finger) over the July column's header. e) Fifth step: keep swiping vertically over the column until reaching the lower section of the column f) Sixth step: in selection mode, place the cursor (using the indicator finger) over the August column's header g) Seventh step: keep swiping vertically over the column until reach the lower section of the column to complete the task.

The general concept is to investigate the advantages of a 3D graphical based query using hand movements over an equivalent 2D interface and also the traditional SQL approaches using keyboard or mouse. In this analysis, both qualitative and quantitative information is obtained over the usability of the interfaces and the general user satisfaction. During this process the 2D and 3D interfaces are compared also with an SQL query using a cube dataset configuration based on the concept of information manipulation, access and retrieval (involving a process of multiple selections and join operations over different related tables). A cube interface configuration is suitable for a hand-gesture based interface, since interfaces of this type resemble aspects of real world interactions. The full interface for these experiments when the task is achieved can be seen in Figure 4.10.



Congratulations, task finished

TASK TO PERFORM:

Using just your hands, select the column ID Code and columns JULY and AUGUST

Column:August

ID	July	August
AB101	71	69
RS102	46	44
GW103	12	12
MD104	87	86

Figure 4.10: Displayed interface when the task is successfully completed.

4.4.3. Evaluation Procedure

During the evaluation process the experiments with the users were divided into of the following steps.

4.4.3.1 Present and explain the experiment and its objectives

The objective of this section is to provide information about the possibility of using 3D hand gesture interfaces instead of the traditional 2D and SQL code based interaction to perform queries on a 3D database. As a result, the proposed experiments perform a comparative study evaluating the users' performance on the proposed prototype interface versus the traditional approaches. Since the users are informed about the overall concept of the experiments and their objectives, we are moving to the next step.

4.4.3.2 Demonstration

The aim of the demonstration section is to present to the users the interface and its elements, answering any related questions. Also, if the user is not familiar with SQL, the basics of the language are explained.

4.4.3.3 Familiarise the subject with the interface

During the familiarisation with the interface stage the interaction mechanisms are presented to the users allowing them to practice with the basic movements and the on screen features.

4.4.3.4 Subject performs the supported actions

The supported functions (e.g. rotate, click, etc.) are explained to the users and demonstrated in real time during this step. Furthermore, they are encouraged to practice and perform these functions by themselves. The total time of training is less than ten minutes.

4.4.3.5 Quantitative data collection

Once the users are familiar with the environment and with the mechanisms to perform the available functions, the full task that was initially introduced is performed measuring the required time to successfully complete it, the number of errors during the users' interaction and also the errors due to the system inaccuracies. Therefore, the quantitative metrics used to evaluate the interfaces for each set of gestures were:

- *the required time to complete the task*
- *the number of errors caused by the users and*

- *the number of errors caused by the system.*

The error metrics (users and system errors) were needed to determinate the influence of them in the general performance; the user errors also provide information about the understanding of the gestures required to work with the interface. The system errors provide information about the correct identification of gestures and the possible improvements required in this technology that could make it sustainable. For each set of gestures, two types of errors were considered during the following actions: clicking in the case of the 3D approach and swiping in the 2D approach.

4.4.3.6 Qualitative evaluation approach

After the interaction task, a questionnaire is completed by the users, evaluating and comparing the available interfaces (i.e. visual 2D/3D and SQL). The questionnaire was the tool to collect users' feedback and to provide a qualitative analysis. During this process any questions from the users are answered to make sure everything is clear to them and additionally there is no time limit for this task.

The model of questionnaire that was used was based on the questionnaires provided by IBM in their research about new interfaces on usability tests. In this case the questionnaire is separated into three main sections: the first two sections evaluate the user's experience with the interface (where section 1 considers aspects related with the interaction process and section 2 aims to evaluate the interface itself) and the third section compares the interfaces and the interactions with a traditional text based SQL approach, as it is shown in Table 4.1. In all questions the user provides a response from 1 to 5 to evaluate the interface, where 1 is the lower score (extremely negative evaluation) and 5 is the maximum score (extremely positive evaluation).

Table 4.1: Usability questionnaire questions.

Section 1	Section 2 - How would you rate:	Section 3- SQL interface compared with the proposed visual approach
Q1: Was the interaction easy to understand?	Q4: The Interface?	Q10: The selection is easier than SQL?
Q2: Was it easy to manipulate?	Q5: The Performance?	Q11: The task is more intuitive than SQL sentences?
Q3: Is the navigation system intuitive?	Q6: The functionality?	Q12: Is it easier to learn the proposed visual approach than SQL?
	Q7: The objective achieved?	Q13: Is the task faster to perform than with SQL?
	Q8: The user experience?	
	Q9: The hand gestures selected?	

Also, at the end of the questionnaire, a last question is asked about the preference of the users over the three approaches (3D hand gesture, 2D hand gesture and typical text based SQL interaction). The scale of evaluation in this case is from 1 to 3, with 1 corresponding to the most preferable approach and 3 to the less preferable one, where the users have to rank the interfaces according their preferences (they cannot evaluate two or more interfaces with the same value).

4.5 Experimental Results

In order to evaluate the proposed interfaces, experiments were conducted using 29 subjects aged between 20 and 50 years old. Regarding the subjects, 63% were males and 37% were female; and 59% had knowledge of SQL and databases. Also the level of programming knowledge and experience was well distributed among all the subjects from novice to expert. The results of the experiments will be separated into two groups: qualitative and quantitative results.

The statistical validation of the data was performed using the Wilcoxon signed-rank test; given the non-normally distributed nature of our data (the level of skewness is too high to be considered normally distributed).

4.5.1. Qualitative results

In this section, the results obtained correspond to the answers given by the users in the questionnaire presented in section 4.4.3.6. The evaluation of the interface from the users is summarised in the following tables and graphs. Table 4.2 shows the median values of the scores for the answered questions in both approaches (with the median absolute deviation value shown in the brackets), in order to avoid the influence of outliers.

Table 4.2: Median values and median absolute deviation in each question for 3D and 2D interaction approaches.

<table>
<tr><th>S1</th><th colspan="2">Q1</th><th colspan="2">Q2</th><th colspan="2">Q3</th></tr>
<tr><td>Median 3D</td><td colspan="2">5.0 (0.0)</td><td colspan="2">4.0 (1.0)</td><td colspan="2">5.0 (0.0)</td></tr>
<tr><td>Median 2D</td><td colspan="2">4.5 (0.0)</td><td colspan="2">3.5 (1.0)</td><td colspan="2">4.5 (0.0)</td></tr>
<tr><th>S2</th><th>Q4</th><th>Q5</th><th>Q6</th><th>Q7</th><th>Q8</th><th>Q9</th></tr>
<tr><td>Median 3D</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td></tr>
<tr><td>Median 2D</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td></tr>
<tr><th>S3</th><th colspan="2">Q10</th><th>Q11</th><th>Q12</th><th colspan="2">Q13</th></tr>
<tr><td>Median 3D</td><td colspan="2">4.0 (1.0)</td><td>4.0 (1.0)</td><td>5.0 (0.0)</td><td colspan="2">4.0 (1.0)</td></tr>
<tr><td>Median 2D</td><td colspan="2">4.0 (1.0)</td><td>4.0 (1.0)</td><td>5.0 (0.0)</td><td colspan="2">4.0 (1.0)</td></tr>
</table>

As we can observe in the first section, the main positive point for the users is related to the first question, which is correlated to the complexity of learning the interaction in both approaches, indicating that the whole mechanism is intuitive and no significant prior knowledge or training is required. The other question with high positive evaluation is related to how much intuitive the system is (question 3), showing that the 3D approach is slightly superior to the 2D one. That can be related to the click movement, which presents a more natural process to select items (as it can be done in the real world). Also, both methods are almost equally evaluated in relation to the manipulation mechanisms (question 2). Furthermore, it can be seen, the interaction mechanisms are considered highly intuitive.

In general, all the questions assessed in section 1 of the questionnaire have values over 3 that indicate a positive evaluation of the interaction process. Also, the 3D gesture based approach has an advantage over the 2D one, indicating that users prefer to interact using 3D hand gestures in a 3D interface. The median absolute deviation over the questions in this section indicates the responses given by the users are in general alike.

In the second section, it can be seen that both approaches have equal evaluation, all over 3.0, indicating both interfaces are well accepted by users.

Section 3 shows a clear advantage for both hand gesture interaction methods over the traditional SQL approach for the defined task (with almost all the questions ranked over 3.5 points), where the aspect with higher score is related to the learning process of the proposed visual approaches over traditional interfaces (question 12). This indicates that the presented visual 3D approaches have a clear learning advantage over traditional methods in database manipulation and interaction.

In general, the results for both approaches are similar. Regarding the obtained values during the evaluation all of them are above 3.5, indicating the general acceptance of the new approaches based on hand movement both for 2D and 3D interaction environments.

Table 4.3: Median values and median absolute deviation (comparison between the 3 approaches), in a scale of 1 (most desirable) to 3 (less desirable).

	Approach 3D	***Approach 2D***	***SQL***
Median	1.5	2.0	3.0
Median absolute deviation	1.0	1.0	0.0

The answers to the last question regarding the users' preferences, comparing the three methods, show the superiority of the hand gesture based methods and particularly the 3D interface over the traditional approaches. Table 4.3 summarises the qualitative preferences of all the subjects (last comparative question) and as it can be seen, the 3D approach is preferred by the users over the 2D approach. This can be explained by the fact that the clicking (3D feature) provides a more stable and intuitive selection mechanism in 3D interfaces.

Regarding the qualitative analysis, the median score values given to all the sections versus different age ranges are shown in Figures 4.11, 4.12 and 4.13. The median absolute deviation is added to the plots.

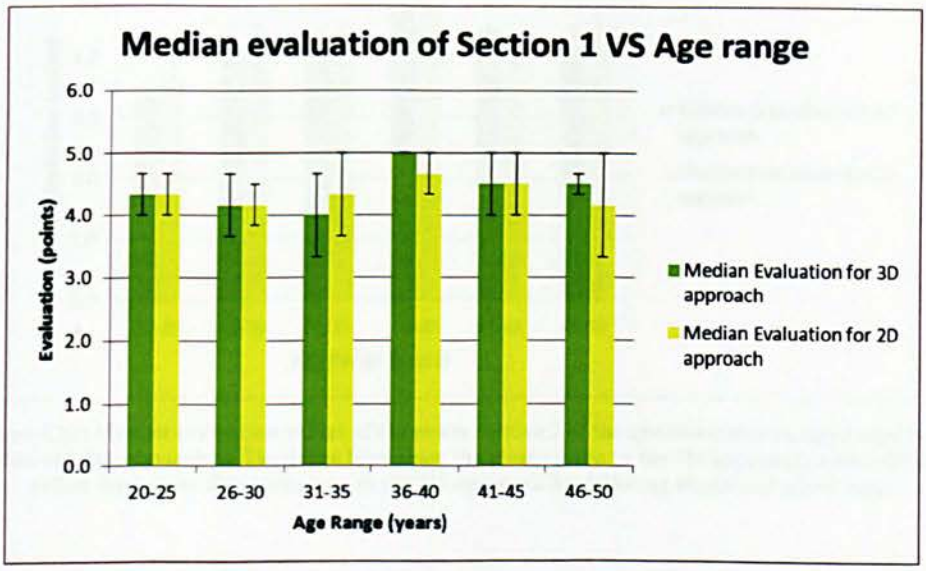


Figure 4.11: Median evaluation values of the section 1 of the questionnaire vs. age range (with median absolute deviation). The green bars show the scores given to the 3D approach; meanwhile the yellow bars show the score given to the 2D approach by different clusters of users' ages.

Figure 4.11 shows the median evaluation scores for all the questions on section 1 (questions one to three) for the different age range of users. The users between 35 and 40 age range gave the highest scores to both hand gesture approaches, but with a clear advantage for the 3D ones. In general, except by the users between 31 to 35 years of age, both interaction approaches received scores over 4 (very positive) indicating that the visual gesture based interfaces provide a desirable way to interact with data.

In the statistical evaluation the first section, the values obtained were *Wilcoxon Statistic* = 133, $p < 0.05$ (one tailed), indicating that the 3D results are statistically significant better than the 2D results for this section.

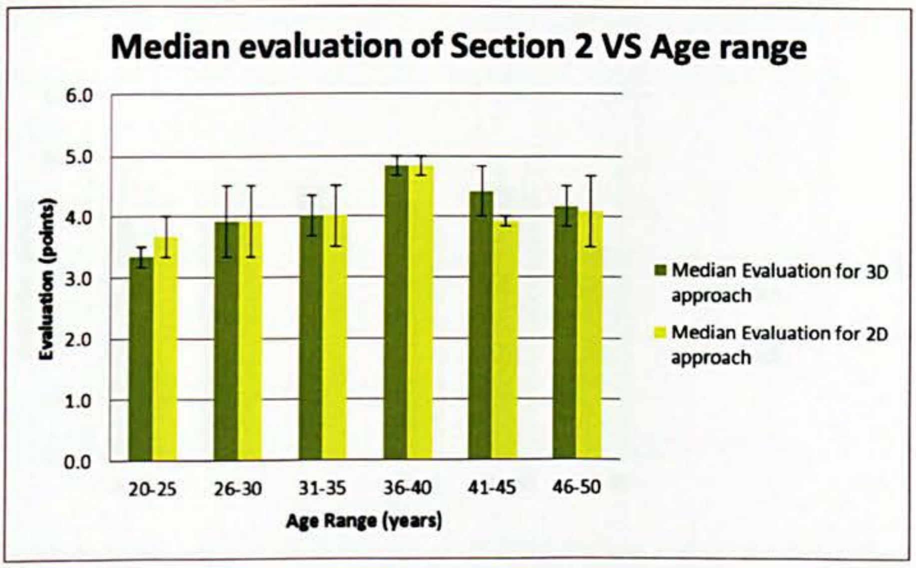


Figure 4.12: Median evaluation values of the entire section 2 of the questionnaire vs. age range (with median absolute deviation). The green bars show the scores given to the 3D approach; meanwhile the yellow bars show the score given to the 2D approach by different clusters of users' ages.

Figure 4.12 shows the median evaluation scores for all the questions on section 2 (questions 4 to 9), and again it can be seen that the group of users between 35 and 40 years of age gave the higher scores for both approaches, but in this case, both of them received the same median scores. In general, an advantage can be seen for the 3D gestures, but the evaluation for both approaches is a bit lower than in section 1. However, it is still higher than 3 indicating a good overall evaluation of the 3D interface.

The values obtained for the statistical evaluation were *Wilcoxon Statistic* = 173, $p > 0.05$ (one tailed), indicating that the 3D results are not statistically significantly better than the 2D results for this section. This indicates the users consider equally acceptable both user graphic interfaces.

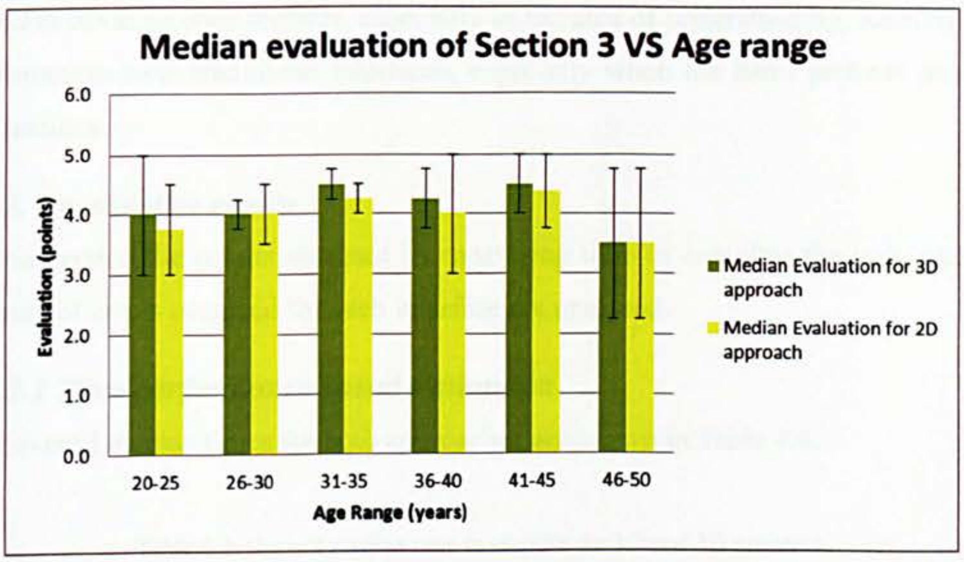


Figure 4.13: Median evaluation values of the entire section 3 of the questionnaire vs. age range (with median absolute deviation). The green bars show the scores given to the 3D approach; meanwhile the yellow bars show the score given to the 2D approach by different clusters of users' ages.

Figure 4.13 shows the results for the section 3 of the questionnaire (e.g. comparison between the hand gesture interfaces and traditional SQL). As it can be seen, the best evaluation was given by users in the range between 41 to 45 years of age, with an advantage for the 3D hand gesture approach. In general, again, the evaluation score is over 3, which indicates a preference of the users for the hand gesture interactions over the traditional text based ones. The lowest evaluation was given by users between 46 and 50 age range, which can be associated to the fact these users have been working generally with text- based interfaces and that would make hand gesture interfaces not that friendly for them, but still desirable. However the high median absolute deviation indicates different opinions between the users of this age range.

In the case of the third section, the values obtained were *Wilcoxon Statistic* = 139, $p > 0.05$ (one tailed), indicating that the 3D results are not statistically significantly better than the 2D results for this section. This indicates the users consider both interfaces preferable to the traditional SQL text based interface similarly.

According to the results presented in the three plots, the significance tests in all sections, regardless of the age range, show the evaluation of the users is highly positive. This means that the hand gesture interaction in a 3D database representation presents advantageous features, especially in the area of understanding, learning and performance over traditional interfaces, especially when the hand gestures provide 3D features.

4.5.2. Quantitative results

In this section the results obtained by measuring time to complete the task, and the amount of errors occurred for each interface are analysed.

4.5.2.1 Time performance based evaluation

The overall median times for both approaches are shown in Table 4.4.

Table 4.4: Overall median time in seconds for 3D and 2D approach.

	3D Approach	**2D Approach**
Median Time (seconds)	22.7	23.8

As seen above, the users in general perform the task faster using the 3D hand gestures. This can be explained by the use of the "3D click" feature, which provides a faster selection of the columns than the swiping, since the rotation feature is the same in both.

Table 4.5 shows the average times for each gender and for users with and without SQL knowledge, highlighting the best time results.

Table 4.5: Median time in seconds and age (in years) for males, females, people who knew SQL and people who did not know it.

	Males	***Females***	***SQL***	***No SQL***
Median Age (years)	32.0	32.5	34.0	29.5
Median TimeAp 3D (seconds)	22.1	**24.2**	**22.1**	23.1
Median TimeAp 2D (seconds)	**21.5**	43.9	29.0	**22.2**

It can be observed, the 3D gesture based interface outperforms the equivalent 2D one requiring less amount of time to complete the tasks in the females and SQL knowers groups. Furthermore, analysing the results over different sub-categories, it can be observed that men perform faster in 2D tasks than females, and also the people who know SQL perform tasks faster in the 3D approach than the people that do not have database programming knowledge. Additionally, there is no significant average age difference over all the available sub groups of users.

The time required to complete the tasks is further analysed providing a more accurate quantitative evaluation. Figures 4.14 to 4.18, presented below, show how different aspects are related with the speed and the time required to accomplishing the tasks.

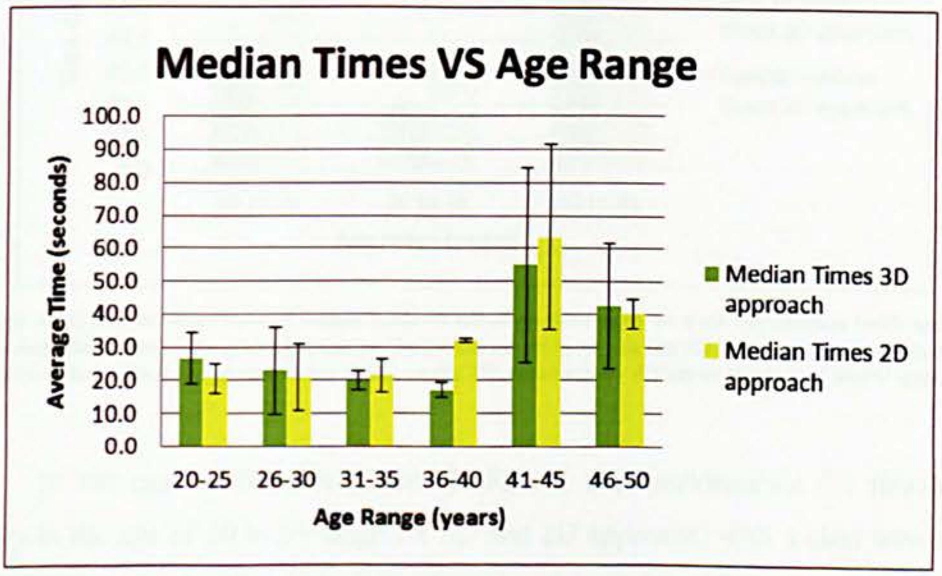


Figure 4.14: Performance median times of all subjects based on their ages for both approaches (with median absolute deviation). The green bars show the performance times on the 3D approach; meanwhile the yellow bars show the performance times on the 2D approach by different clusters of users' ages.

In figure 4.14 it can be seen that subjects between 26 and 40 age range have the best performance in time, especially in the 3D approach, while the slowest performance is for the subjects aged 41 to 45 range. Also, the 3D approach in general has better time performance than the 2D one. This can be explained by the previous use of graphical interfaces and other more interactive technologies, which would allow a better understanding of 3D interfaces and gesture interaction.

Users between 41 to 45 age range have the slowest time results, yet the highest median absolute deviation, which indicates high variations between the time performances between the users of this age group. In the opposite case, the users that have the best times (31 to 35 years old) have low median absolute deviation, indicating that the users with better results have a general good understanding and performance, which can be explained by a different level of knowledge on the use of interactive touch or gesture technologies.

Figures 4.15 and 4.16 present the results obtained by males and females respectively.

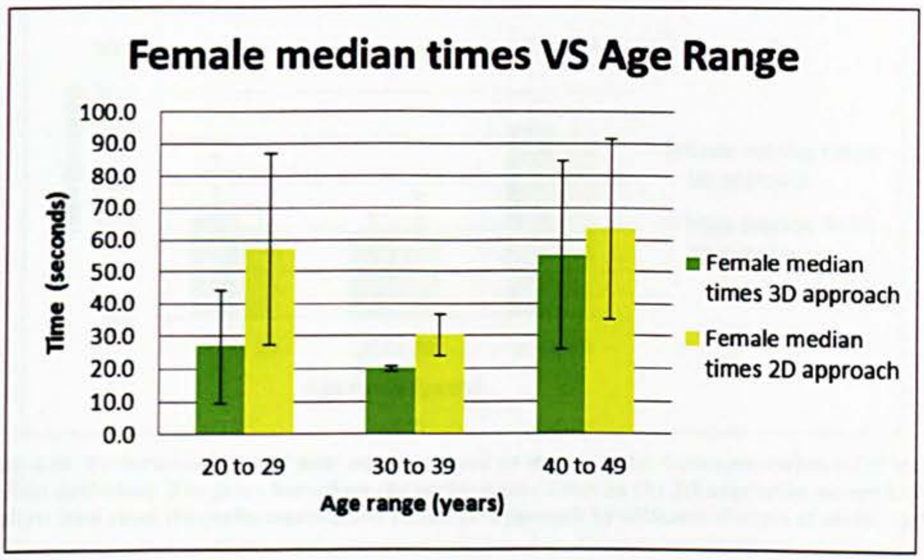


Figure 4.15: Performance time of female subjects based on their ages for both approaches (with median absolute deviation). The green bars show the performance times on the 3D approach; meanwhile the yellow bars show the performance times on the 2D approach by different clusters of users' ages.

In the case of females (figure 4.15), the best performance for females is clearly in the age of 30 to 39 range for 3D and 2D approach, with a clear advantage for the 3D approach. That indicates a better understanding of the functionality and how to perform the different gestures, related with the clicking feature. It is also clear, that at this age range, the median absolute deviation of task's performance times is the lowest, which indicates the results in general are similar, especially for the 3D approach. For all age range, the 3D approach has better performance, which indicates the selection using clicking provides faster results than the swiping selection.

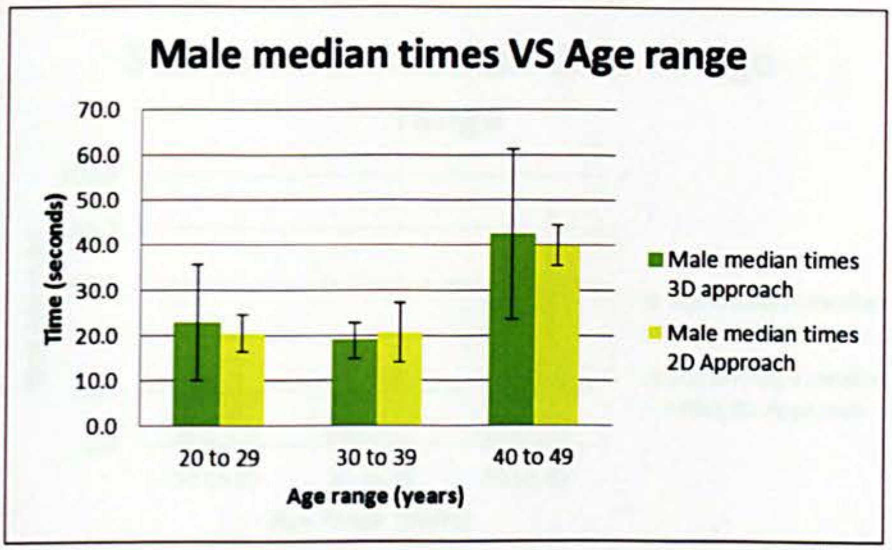


Figure 4.16: Performance time of male subjects based on their ages for both approaches (with median absolute deviation). The green bars show the performance times on the 3D approach; meanwhile the yellow bars show the performance times on the 2D approach by different clusters of users' ages.

For male subjects (Figure 4.16), the best result for the 3D and 2D approaches is in the range of 30 to 39 years of age, with an advantage for the 3D approach (the same that happened with females) indicating that in general terms that age range have a better understanding of the use of 3D interfaces based on hand gestures. Also, the median absolute deviation for both approaches in this age range is low, indicating a low difference between the results obtained by each male user. In general, the 2D approach has better results, but the deviation of the data for the 3D approach in the range of 20 to 29 and 40 to 49 years of age is high, indicating some users in that age range may have more experience on this kind of interfaces than others.

According these results, it can be concluded that gender is not related to the performance in the group with best results (30 to 39 years old), since the results show the same tendency.

In figures 4.17 and 4.18 the time results for users with and without SQL knowledge are presented.

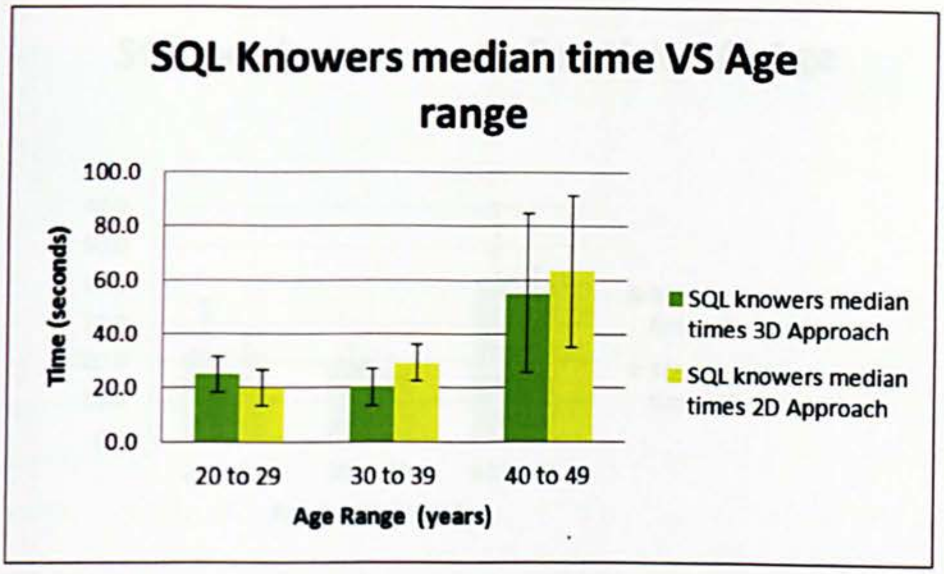


Figure 4.17: Performance times of subjects with knowledge of SQL for different age ranges (with median absolute deviation). The green bars show the performance times on the 3D approach; meanwhile the yellow bars show the performance times on the 2D approach by different clusters of users' ages.

For users with SQL knowledge (Figure 4.17), the best results in time performance using the 3D approach were obtained by those between 30 and 39 age range, while in the case of the 2D approach, the users between 20 and 29 years of age had the best results. In both ranges, the median absolute deviationmedian absolute deviation was relatively low; indicating the performance of the users and that the level of understanding of the gesture interaction is similar for all the users in these age ranges. In the case of users in the range from 40 to 49 age, where the slowest results for both approaches were obtained, the 3D approach has an advantage, but for both of them the median absolute deviation is high, which indicates the interaction was better understood for some users, which can be related to their previous use of similar interfaces.

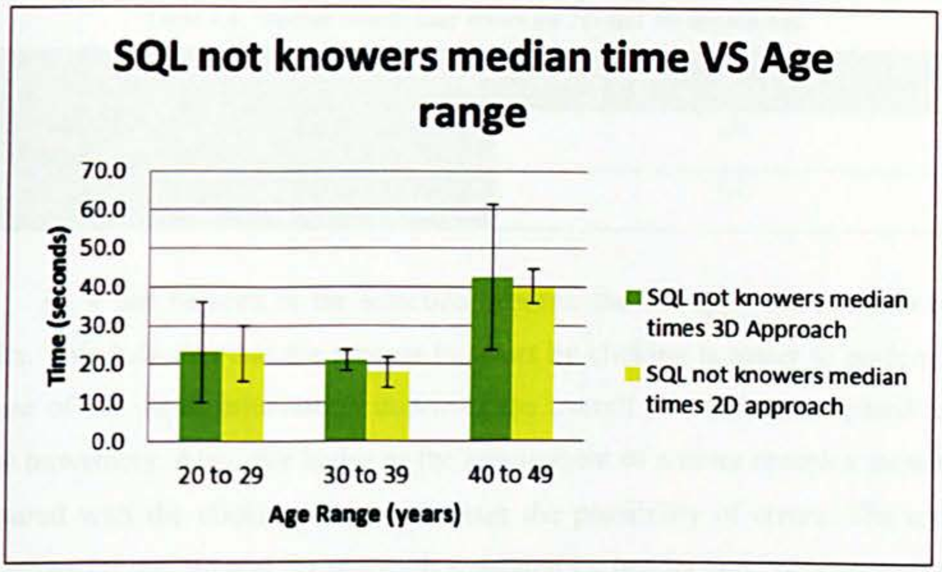


Figure 4.18: Performance times of subjects with no knowledge of SQL for different age ranges (with median absolute deviation). The green bars show the performance times of the 3D approach; meanwhile the yellow bars show the performance times on the 2D approach by different clusters of users' ages.

In the case of users without SQL knowledge (Figure 4.18), the best results for both approaches are obtained by the users between 30 and 39 years old, with a clear advantage for the 2D approach. Also, this range of users presented lower median absolute deviation values in both cases. This advantage for the 2D gestures indicates users without SQL knowledge understand better swiping over clicking to perform a selection for the average of users in this age range, possibly related to previous use of similar technologies or better understanding of the interaction process.

Comparing the times obtained in both approaches, the statistical evaluation values obtained were *Wilcoxon Statistic* = 135, $p < 0.05$ (one tailed), indicating that the 3D results are statistically significant better than the 2D results.

4.5.2.2 User errors

In this section, the selection gestures are analysed. The overall median number of users errors during the performance of the task for the 3D and 2D approaches are shown in Table 4.6.

Table 4.6: Median overall user errors for 2D and 3D approaches.

		Median Errors
Median Error Selecting (amount)	**3D (by clicking)**	1.0
	2D (by swiping)	1.2

As it can be seen in the selection process, the 3D approach presents better results. This indicates that the process to select by clicking is easier to perform and the use of the depth information improves the overall interaction compared to the swipe movement. Also, this is due to the requirement of a more complex movement compared with the clicking which increases the possibility of errors. The median user errors for the 3D and 2D approach separated by gender and knowledge of SQL are shown in Table 4.7.

Table 4.7: Median amount of user errors for males, females, people who knew SQL and people who did not know it for 3D and 2D approaches.

2D and 3D Gesture approach Errors					
		Male	***Female***	***SQL***	***No SQL***
Median Error Selecting (amount)	3D (by clicking)	1.0	1.5	1.0	1.0
	2D (by swiping)	1.0	2.0	1.5	1.5

Table 4.7 clearly demonstrates female users made more mistakes using the swiping gesture (2D) than the clicking one (3D). Comparing with the results in table 4.5 (median times), it can be said there is a direct relation between user errors and the time taken to perform the tasks.

For users with SQL knowledge, regarding the selection, both groups of users had better performance in the 3D approach. These results (compared with the ones in table 4.5) reinforce the theory of a direct relation between user errors and time performance.

In order to understand the users' behaviour and the distribution of their errors, the total errors according to their age ranges are displayed in Figure 4.19.

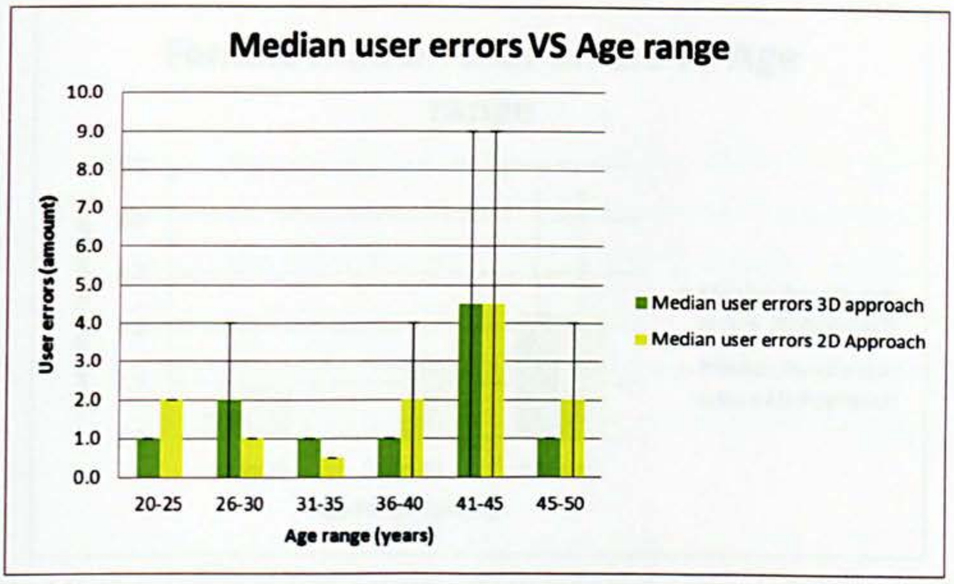


Figure 4.19: Median user errors according the age range (with median absolute deviation). The green bars show the amount of user errors on the 3D approach; meanwhile the yellow bars show the amount of user errors on the 2D approach by different clusters of users' ages.

Analysing the median user errors in each approach, presented in Figure 4.19, it can be seen that the worst performance was achieved by the group between 40-45 years old, but with a notably high median absolute deviation, showing that the median is not a clear indicator of the performance of the users and that some of them were capable of using the interface fairly easily. The best results for the 3D set of gestures can be seen in the ranges between 20 to 25, 31 to 40 and 45 to 50 years old, while the 2D interaction has fewer errors for users between 31 and 35 years old, but with more uneven results in general. These good performances on the 3D approach can be related to a better understanding of the clicking gesture.

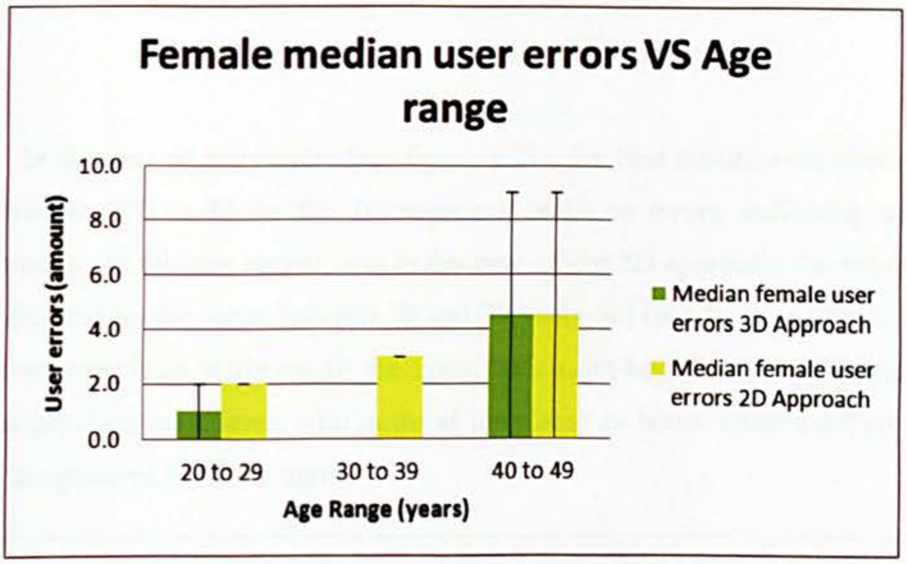


Figure 4.20: Median female user errors according the age range (with median absolute deviation). The green bars show the amount of female user errors on the 3D approach; meanwhile the yellow bars show the amount of female user errors on the 2D approach by different clusters of users' ages.

Female users made fewer errors on 3D gestures in the age range between 30 and 39 years of age, while female users between 20 and 29 years old had fewer mistakes in the 2D interface (see figure 4.20). Also, the users between 40 and 49 age range present the highest median number of errors, but with high median absolute deviation, indicating that some users had a better understanding of the interface and manipulation than others, possibly related to previous use of gesture based technology.

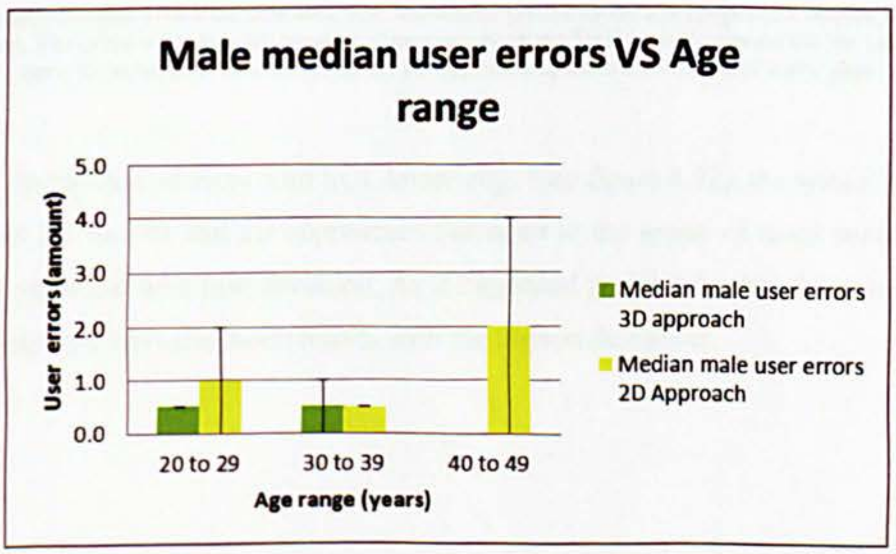


Figure 4.21: Median male user errors according the age range (with median absolute deviation). The green bars show the amount of male user errors on the 3D approach; meanwhile the yellow bars show the amount of male user errors on the 2D approach by different clusters of users' ages.

In the case of male users (see figure 4.21), the best results were obtained by users between 40 to 49 for the 3D approach (with no errors, indicating accurate performance for this age range), and in the case of the 2D approach, the best results were obtained by the users between 30 and 39 years old (see figure 4.21). Also, the deviations were high in almost all the cases, which can be related to several factors, such as previous experience with gestural interfaces or better understanding in the use of the gestures for some users.

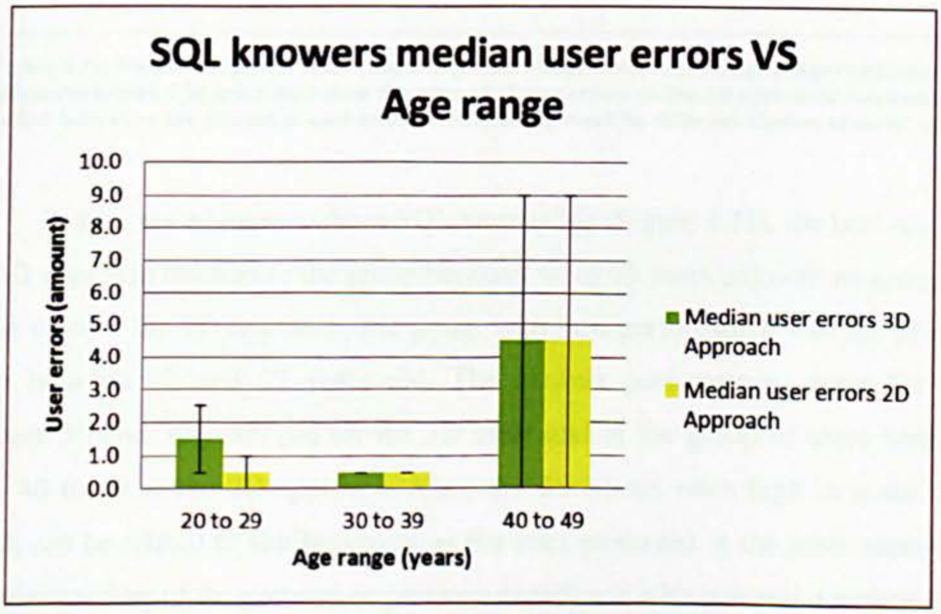


Figure 4.22: Median errors for user with SQL knowledge according the age range (with median absolute deviation). The green bars show the amount of user errors on the 3D approach; meanwhile the yellow bars show the amount of user errors on the 2D approach by different clusters of users' ages.

In the case of users with SQL knowledge (see figure 4.22), the lowest number of errors for the 3D and 2D approaches belonged to the group of users between 30 and 39 years old with low deviation. As it happened previously, the oldest users- 40 to 49 years old have the worst results with the highest deviation.

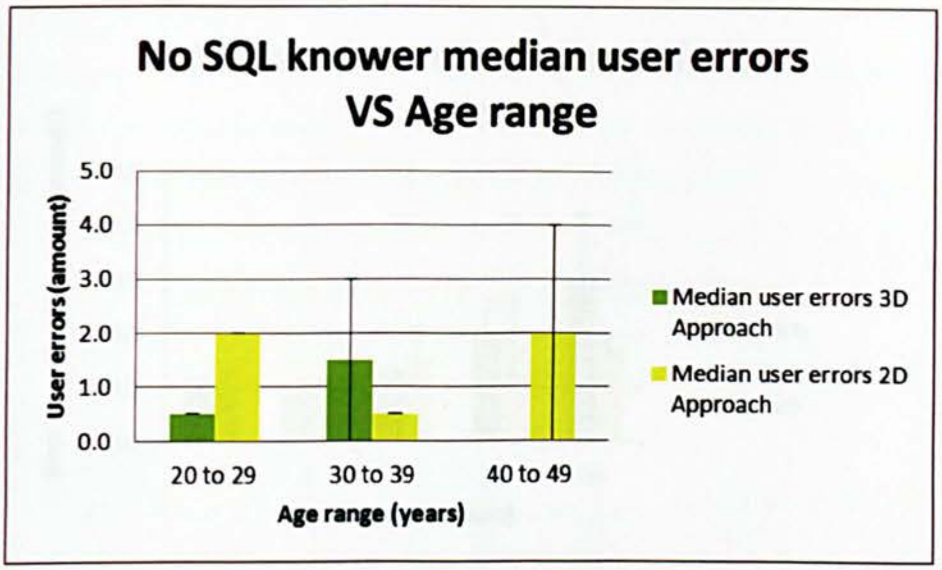


Figure 4.23: Median errors for user without SQL knowledge according the age range (with median absolute deviation). The green bars show the amount of user errors on the 3D approach; meanwhile the yellow bars show the amount of user errors on the 2D approach by different clusters of users' ages.

In the case of users without SQL knowledge (Figure 4.23), the best result for the 3D approach belongs to the group between 40 to 49 years old with no errors, and in the case of the 2D approach, the group with best performance was the group of users between 30 and 39 years old. The slowest performances were for users between 30 and 39 years old for the 3D case; and in the group of users with ages from 40 to 49 in the 2D approach. Also, the deviations were high in some cases, which can be related to similar causes as the ones presented in the other cases (such as understanding of the gestures or previous experience with gestural interfaces).

In general terms, the previous figures show advantages for the 3D approach, especially in the case of the users with the best performance (30 to 39 years old). Also, the previous mentioned group presents in general the lowest deviation, indicating the consistency with the results presented in the section 4.5.2.

Finally, to evaluate the influence of the users errors on performance, in figure 4.24 a comparison considering the number of errors is shown. In general, the number of errors was less than 3 (see Table 4.8). Also, it can be seen a correlation between the number of errors and the time required to accomplish the task.

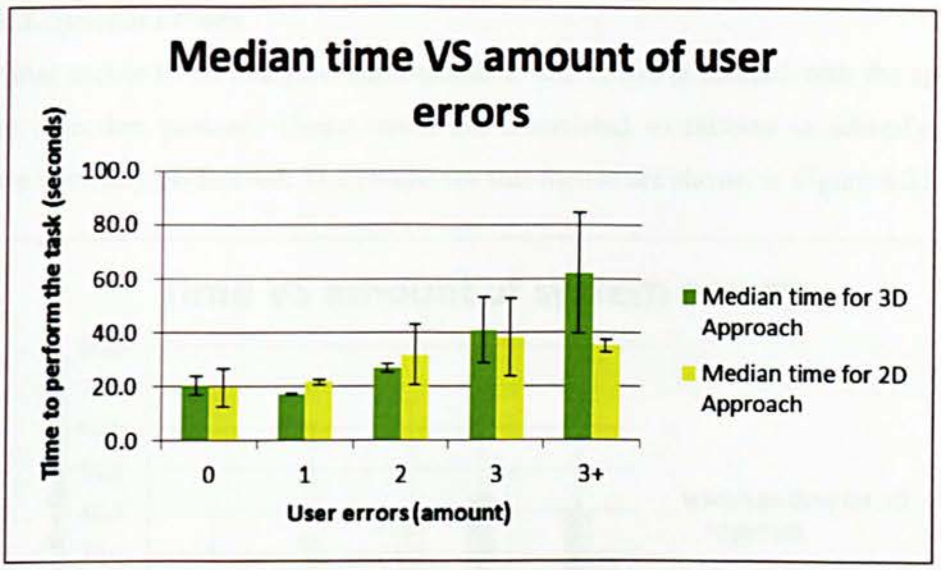


Figure 4.24: Median time according the amount of user errors (with median absolute deviation). The green bars show the time on the 3D approach; meanwhile the yellow bars show the time on the 2D approach by different clusters of amount of user errors.

Table 4.8 Amount users according amount of user errors.

		Amount of user errors				
		0	**1**	**2**	**3**	**3+**
Amount of users	**3D approach**	17	3	3	3	3
	2D approach	15	3	5	2	4

The general conclusion in this section presenting a new 3D interaction approach is that there is a clear correlation between the number of users errors and the time required to perform the task, but this is not always clear in the case of the 2D approach. Also, it has been shown that a 2D approach is more prone to errors and the use of a third dimension can help to improve functionality and reduce the number of user errors during the execution of tasks in a 3D visual environment.

Comparing the amount of user errors obtained in both approaches, the Wilcoxon signed-rank test values obtained were *Wilcoxon Statistic* = 122, $p < 0.05$ (one tailed), indicating that the 3D results are statistically significant better than the 2D results for the amount of user errors during the execution of the evaluation task.

4.5.2.3. System Errors

The final metric to be analysed corresponds to the errors generated with the system in the selection process. These errors are associated to failures in identifying a gesture correctly performed. The results for this metric are shown in Figure 4.25.

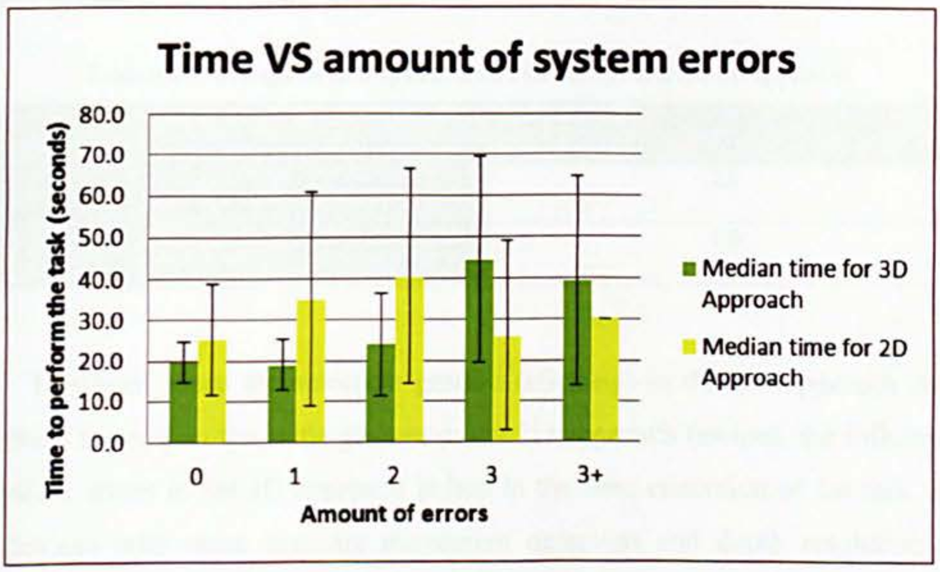


Figure 4.25: Median time according the amount of system errors (with median absolute deviation). The green bars show the time on the 3D approach; meanwhile the yellow bars show the time on the 2D approach by different clusters of amount of system errors.

As seen in Figure 4.25, there is a correlation between the number of system errors and the time to complete the tasks in the 3D approach. However, that is not clear in the case of the 2D interactions. Also, the number of errors in identification 3D and 2D gestures is concentrated between 0 and 2 (as it can be seen on Table 4.9). Figure 4.25 also shows that the 3D approach presents in general lower deviation than the 2D one, especially in the cases of fewer errors.

Table 4.9: Amount users according amount of system errors.

		Amount of system errors				
		0	**1**	**2**	**3**	**3+**
Amount of users	**3D approach**	8	7	4	2	6
	2D approach	13	6	6	3	1

Table 4.9 shows that the occurrence of system errors tends to be less than 2 for the 3D and 2D approach, but, in the case of the 3D approach, there are more users with more than three system errors than in the case of 2D. Also, the total median amount of errors (Table 4.10) shows a similar result, where the 2D approach has less system errors then the 3D one.

Table 4.10: Average overall system errors for the 2D and the 3D approach.

		Average Errors
Median. Error Selecting (amount)	**3D (by clicking)**	2.0
	2D (by swiping)	1.0

However, since the selection gesture (clicking) in the 3D approach can be performed faster than the same gesture in the 2D approach (swipe), the influence of the system errors in the 3D approach is less in the time execution of the task. Also, new devices with more accurate movement detection and depth resolution (e.g. Kinect 2) could reduce these types of system errors, which can improve significantly the results in a 3D gesture interface.

Comparing the amount of system errors obtained in both approaches, the values obtained for the statistical evaluation were *Wilcoxon Statistic* = 77, $p < 0.05$ (one tailed), indicating that the 3D approach's results are statistically significant better than the 2D approach for the amount of system errors during the execution of the evaluation task.

Table 4.11 shows the result obtained for the Wilcoxon statistical evaluation test for each of our experiments.

Table 4.11: Statistical significance values for the experimental data obtained.

Experimental Data	**Amount of subjects involved (N)**	**Wilcoxon Statistical**	**P-Value**
Qualitative questionnaire Section 1	29	133	0.04182
Qualitative questionnaire Section 2	29	173	0.35197
Qualitative questionnaire Section 3	29	156	0.14231
Average performance time	29	135	0.03515
Average amount of user errors	29	122	0.03144
Average amount of system errors	29	77	0.00114

4.5.2.4. Qualitative and quantitative results' comparison

The comparison between the qualitative evaluation given by the users and quantitative results obtained by them (times and errors) are presented in this section.

The case of the qualitative results presents some interesting results. Figures 4.11 and 4.12, demonstrate users between 35 and 40 years of age give the best evaluation to both approaches to do with aspects of interaction and interface (sections 1 and 2 of the questionnaire respectively), with an advantage to the 3D approach. In the case of the comparison with a traditional SQL interface, the best evaluation to both approaches was given by the users between 41 and 45 age range. In the three graphs presented, there is not a clear correlation between age and preference over one specific approach, but it is clear that all the evaluations in the three sections are positive.

Figure 4.14, points out that the best time results for users' average time performance were obtained by those users in the age range between 30 and 40 years old. In the case of the user errors (Figure 4.19), there is a similar result for the group with the highest number of errors, but that tendency is not that clear in the case of the best performances. This fact can be related to the users' speed to perform the gestures and the general task defined. In general terms it can be said the users with best performances are those with ages between 20 and 40 years old.

Comparing qualitative and quantitative results, there is not a clear relation between preferences by the users (qualitative evaluation) and their performances. This can be related to the novelty of the interaction approaches (especially in the case of the 3D approach) that makes the users feel comfortable with the interface, regarding the time required to perform the tasks or the errors committed during the interaction process..

4.6. Conclusions

In this chapter a 3D gesture based interface for 3D database interaction was presented. This interface is based on hand gesture interaction in 3D environments providing a more natural interaction to the end user. Furthermore, this approach incorporates different interaction methodologies, which were analysed providing all the details of their mechanics, allowing the definition of more complex interaction systems for data manipulation in future work.

To evaluate and validate this framework, two sets of interaction experiments were performed, using Microsoft Kinect to capture the hand gestures. The experiments indicate differences over the users' experience for the two models that were introduced (2D and 3D hand gesture interaction) operating both under the same interface. The experiments show a clear preference for the hand gesture interaction over the traditional keyboard and mouse based interfaces, and especially the 3D based approach. In general these interfaces operated by the users' hands, are regarded as a more suitable and effective input methods mainly for 3D than 2D tasks. Finally, it can be concluded that the proposed 3D hand gesture interface is more intuitive and less amount of time and training effort are required to understand and apply it on different tasks and mainly on 3D databases.

The range of ages that presented better results for all cases is between 20 and 40 years old, who required lower amount of time to perform the tasks resulting in a lower number of errors in general.

Additionally, there is a correlation between the amount of user errors and system errors with the time required to perform the tasks in the case of the 3D set of gestures. However, the number of errors in general is very low and does not affect significantly the users' performance and the users' satisfaction.

It can be argued that after the experiments presented in this chapter, 3D interfaces based on hand gestures improve the users' experience, reducing the required learning time and the overall task procedure. Also, the application of these interfaces in data manipulation systems indicate a better understanding of the tasks by the users, especially when the hand gesture interactions support 3D gestures, providing a more natural and less complex gestures to interact with a system, simplifying the overall interaction process.

This chapter presented a 3D approach to interact with databases in a 3D environment, aiming to improve the developer's interaction. But there is still a need to extend this kind of technology for other software development areas. Since the application development process for 3D interactive interfaces is better achieved in a 3D environment discussed on section 2.5, especially for multi-threaded software development, experiments on that area are highly desirable. Chapter 5 introduces a framework for multi-threaded programming and a set of experiments to validate its features over a traditional 2D programming interface.

Chapter 5

HCI for multi core programing

5.1. Context and Overview

In chapter 4 we presented a method to interact with 3D representations of databases using direct 3D touchless interaction, removing intermediate devices, such as tablet PCs or mobile touch devices. The removal of intermediate devices helps to achieve a more natural interaction, especially when aimed at developers. However, there is a need to overcome the 3D development problems in a lower level: programming interfaces. The first aspect to explore is the 3D user interaction.

Nowadays, three-dimensional graphical interaction is a common element of many applications. Attempts to provide a bridge between the real environment and the computer interfaces have become an important topic in human computer interaction research. The significance of these Human-Computer interaction challenges have been addressed by several researchers in the last few years, highlighting the importance of creating new communication methods between humans and computers, replacing traditional methods and devices (Fishkin, 2004).

Video game technologies are leading the way on generating interactions using body commands of the users, especially hand gesture based interactions. This is due to the flexibility and the intuitive use of the hands during the interaction and the manipulation of 3D objects in 3D space. In many cases, simply having two dimensional interactions is not enough to perform specific tasks naturally, especially when these activities are performed in three dimensions in the real world. Advances in depth capturing devices have provided novel approaches to interface systems, as Microsoft Kinect has shown lately (Guettard et al., 2011). This data interaction mechanism can be extended to have real impact on the computer and software development community.

A novel research area for 3D hand gesture interaction is multi-thread programming, since the representation of multiple lines of code executed simultaneously can be better represented and understood in a three dimensional graphic environment than with simple sequential code, because of the resemblance with the real environment (Chau et al., 2013; Wagner et al., 2013). The complexity of generating applications using multiple threads lays generally in the lack of representation of the final program and how it will operate. The issues related to the working environments in developing applications for multiple processors/threads are not new and there have been advances, indicating the need for novel and advanced interactive mechanisms (Wang et al., 2007; Ryoo et al., 2008; Nickolls et al., 2008). Using graphic icons to represent data elements and functions helps to clarify their purpose in programming, and environments have successfully represented these tools in a fairly accurate and intuitive way (Dolbeau et al., 2007). Most of them lack 3D representation, which, as indicated, increases the understanding of the encapsulated information and the productivity during the utilisation of multiple sources of information, especially in complex tasks such as multi-thread programming (Kumar and Benbasat, 2004).

In this chapter we present a novel approach for generating multi-thread code using hand gesture interactions in a 3D environment, introducing the concept of multidimensional software programming and design. Among the other advantages of 3D software development, the proposed framework allows the user to navigate in a more human-friendly code development environment, while the proposed human computer interaction mechanism takes advantage of all the features and concepts of 3D interaction systems. In the following section, an analysis of previous work is presented, showing the progress in the related areas. Then, the proposed environment is analysed providing details of novel interaction framework for software development. Finally, an evaluation methodology of the proposed interaction approach is discussed and conclusions are presented. The contributions related to this chapter are the definition and implementation of an interaction interface for multi-threaded programming, 3D novel representation of iconic tools and the introduction of a 3D representation for multi-thread programming.

5.2. Previous Work

New interface paradigms have made clear the need for improvements in the way we interact with information. The work of MIT's Tangible Media Group (Ratti et al., 2004) presents an alternative to replacing the text-driven systems in geographic information systems (GIS). Their approach allows direct interaction with geographical data, where the user can modify and analyse surfaces as part of the interface using tangible objects, (such as blocks, trees, hills, etc.), integrated with augmented reality environments, depicting changes of specific terrain characteristics. A 3D display provides for the user, a visualisation of the work in progress, using the "tangible bits" paradigm and digital elevation models of a surface. The Tangible Bits paradigm (Ishii and Ulmer, 1997) consists of an augmented reality system combined with an intelligent environment, where users can manipulate real objects on a surface and obtain feedback from the interaction surface based on digital projection. The system uses laser-based technology to detect the movements of the user and advanced image processing software based on augmented reality techniques to generate feedback. The main problem of implementing this kind of interface is the high cost of the devices used and the difficulty to configure all the hardware and software for a single application. The interaction mechanism was also limited to the particular problem without providing any flexibility. These types of advances show the possibility of using 3D interaction scenarios in other kinds of information manipulation, such as programming.

Finding an adequate way to design a system with 3D interaction and natural interface is a complex problem because of the lack of information regarding the working area and the needs of the user. Paradigms like case based reasoning and the use of support frameworks to design new object-oriented architectures presented in the work of Vazquez (Vazquez et al., 2010) can help to overcome the previously mentioned difficulties. The importance of taking into account previous designs and being able to use them in the process of materialisation of new software is discussed. Based on this approach, the test system is able to provide "advices" to developers on the choice of architectural software components. These advices are based on the evaluation of a set of quality aspects, for example performance, modifiability or scope. As a consequence, experience is required to decide if the components are really suitable for software design.

The possibility of representing in 3D a large amount of information was explored exhaustively in the work of Marcus (Marcus et al., 2003) where 3D models were used to represent a software system allowing a better understanding of high dimensional data. The most relevant aspect of this system is related to user interaction and 3D visualisation, showing multiple nested levels of the code, based on colours and viewing models. Probably the major drawback of this design is that the interaction is in 2D and that it's based on traditional interaction devices. Consequently, it retains the disadvantages of a 2D interaction in a 3D environment, with the interaction mechanisms limited with this model of representation.

There have been many attempts to achieve a pure 3D programming language over the years. A relatively modern example is the Solid Agents in Motion –SAM- (Geiger et al, 1998), which is a visual 3D programming language for parallel systems and animation. The language is based on agents (3D objects with an arbitrary number of input and output ports) that interact by exchanging messages (a data structure that can be an identifier, a value, identifiers of the sender or receiver enclosed as text). The behaviour of each agent is specified by production rules with a condition and a sequence of associated actions, as in state machines. The graphical representation of each element is initially a semi-transparent 3D model, such as cylinders, spheres or cones. Agents and messages have an abstract and a solid 3D representation, where the abstract representation consists of a description text based on the agent's action environment, the agent itself and the production rules applicable to the agent, while the solid 3D representation corresponds to a graphic 3D model of the agent. The interaction with these elements (mouse and keyboard based interaction) can be achieved by moving over the 3D object representation, and, getting more specific information, by double clicking over the object. Each 3D representation element has a number of connection ports for data input and output depending on the definition and function of the agent. The agents use these ports to send and retrieve messages and each port has textual identifiers and colours to indicate their function. In Figure 5.1 an example of an agent can be seen. The execution of the programs based on these graphic agents is based on synchronous communication, in a cycle of two phases: i) agent execution and ii) agent communication. In the first step, all the agents check their execution rules and proceed to perform their tasks in the respective order and then pass the message to the next agent, according to their execution rules. Even when this model of programming presents a lot of possible advantages, the complexity of the rule generation and the trivial non-natural methods of interaction (using mouse clicking) present a poor use of 3D interaction capabilities, especially in the aspect of visualisation and interaction with the programming environment.

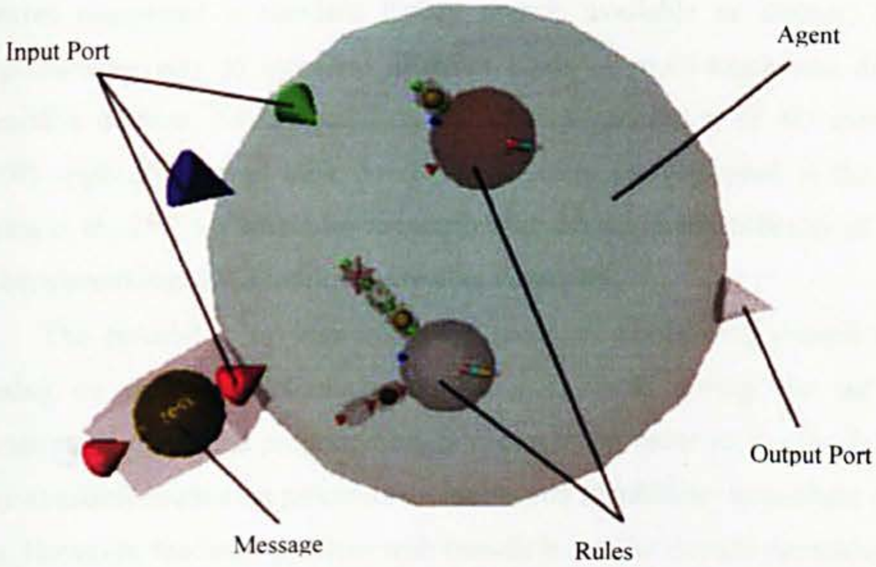


Figure 5.1: SAM's agent architecture (Geiger et al., 1998).

Later on, attempts on 3D graphic interaction proposed animated execution of programs instead of text based debugging. One example of this is the approach presented by 3D-PP, a visual programming system with 3D representation (Okamura et al., 2004), where the construction of programs is based on a hierarchical graph of nodes and edges, where the nodes correspond to data (represented by spheres), operators (represented by inverted cones) and process (represented by pillars). Each process is defined by a set of rules, composed by a condition (used to select one rule from multiple choices at runtime) and a body (that defines the performance rule following the state machine model). All the components can be accessed and modified using direct input devices, such as mouse or any pointing mechanism. Once the program starts, there are multiple options, including stopping the animation and modifying the program to solve possible problems and bugs. The main issue of this implementation lies on the fact that all the construction graphical elements work as a normal 2D code based program, where the process of execution and visualisation needs to be defined step by step. Also, the interaction is still based on traditional interfaces, not using the available features and advantages of 3D interaction and manipulation, such as rotations, multiple angle view of the software, zooming, etc.

In (Ren and O'Neill, 2012) the usability of freehand gestural target selection with different 3D marking menu layouts and target directions were analysed. Gestures comprised a standard library that is available as default, offering a comprehensive way to integrate different kinds of multi-touch and direct input interaction devices. More details on the current generation of 3D user interface (3DUI) applications, and their development issues are presented in the survey in (Takala et al., 2012). Criteria for measuring the development difficulty of 3DUI and two benchmarking 3DUI toolkits were also suggested.

The possibility of improving the speed of performing complicated tasks focusing on their parallelization has being explored during the lastest years. Nowadays, multi-thread programming is necessary in order to use the full capacity of the available multi core processors with built-in capabilities to perform concurrent tasks. However, the main problem with threads is not the threads themselves, but the lack of visualisation of related components during the design and programming processes that cause several problems related to concurrency, previously mentioned in chapter 2 (section 2.1.3).

Another issue related to multi-threading is the lack of standardisation of the programming 2models for parallel computing in heterogeneous systems. The MERGE framework presented by Linderman (Linderman, 2008) addresses this problem by considering an intense use of "libraries" to deal with tasks and the data distribution between the different components of heterogeneous systems, using a unified programming approach instead of the classic static/dynamic compilation method. This dynamic approach allows the programmer to designate specific tasks to architectures without knowing exactly which machine is going to be carrying out each task. The framework uses knowledge of the architecture (based on a set of libraries) to distribute the work between the components of the system. This approach addresses several issues related to the parallel programming in multiple machines, but is not a real solution when parallel computing is done in just one of the processors and task assignment is defined manually, which generates the problem of lack of visibility of the different components in the application. Under this perspective, the use of a graphic-based environment becomes more suitable to solve programming issues of multi-core processors.

The approach for multi threads presented by Harrow (Harrow, 2000) aimed at the need of visualisation of concurrent executions and provides an approach to model the working threads in real time and their progress in the system, but still uses as a starting point written code, where learning time is higher and therefore errors are more probable because of the complexity of the elements to understand, learn and use. Also, it does not provide multiple view options, because the threads' tasks overlap. The availability of a 3D graphic framework to visualise how the program is constructed becomes more desirable and intuitive.

The combination of a 3D visual interface and a 3D gesture based interaction system seems to be an attractive and interesting way to solve the problems previously discussed about concurrent programming. The proposed approach of a novel framework is presented in the following section.

5.3. Methodology for multi-threaded 3D programming

The multi-thread framework proposed is divided into different key elements related mainly to the interaction capabilities and the interface mechanisms.

At this stage, it is necessary to explain the importance of using this approach in a programming environment before analysing the framework. The use of 3D interaction is possibly the most arguable issue in this work, but there are several reasons related to previous works in the area of human-computer interaction that support this approach. In more detail, there are several studies on designing systems that can support 3D interactions, increasing their capabilities including natural gestures and providing to the users more confidence and comfort during the interaction process. The advances in interfaces are aiming to have more "human friendly" interactive systems, where the traditional external devices are getting replaced by novel interaction mechanisms. Particularly, the advances in infrared motion capturing devices, which provide the possibility of interacting directly with systems using bare hands, have been used successfully to achieve more natural interaction. These devices, which have been extensively used in entertainment, are gradually entering other research and development areas, especially those related to hand gesture based interaction frameworks. This is due to the simplicity of understanding interactions that are part of everyday human activity, such as grabbing and moving objects in a given 3D space. These actions and gestures can be easily applied to interfaces related to graphic-based programming (MacLaurin, 2011; Karpak et al., 2011). The idea to use 3D metaphors to represent a program is not new, especially in robot programming, where several tasks are performed in real environments and an iconic visualisation is utilized to simplify the definition of graphic elements for specific tasks (Biggs and MacDonald, 2003). These ideas are also applicable to multi objective linear programming (i.e. programming algorithms to solve optimization problems for multiple variables that can use parallel programming to solve the mentioned problems) (Baky, 2010; Karpak et al., 2011) and 3D programming for dynamic systems (i.e. systems that change during time, such as particle movements, bioreactors, communication systems, etc.). Consequently, we might argue that there is an interesting possibility to use the proposed methodology for multi-thread programming.

5.3.1. Interaction definition for a 3D gesture-based programming environment.

The system interaction is based on direct hand instead of hand and fingers gestures as in the interface presented in chapter 4. In this approach, the skeleton tracking and palm 3D detection provided by the Microsoft Kinect SDK is used to perform the interactions.

Since the interaction with the system is directly based on 3D hand gestures and using 3D objects as metaphors, three basic interactions are defined:

- **Rotation:** Rotations in the system are based on simply moving the hand in the defined working area. This action only works in the workspace of the framework and its main utility is to shift between the different available threads of the program. The rotation of the threads is around the vertical axis, moving from left to right or vice versa with a predefined 3D space assigned to each thread. The action is performed by sliding the hand over the working area making it possible to perform it either from left to right or from right to left. It is also possible to perform it only if there are no other actions or gestures performed at that time by the user.
- **Grab:** The grabbing process is required to select and add programming elements into a given thread. These elements are in the programming tools area of the screen (Figure 5.2), out of the workspace. The grabbing process is done in two steps: first, hovering over the selected item and then moving the element to confirm the process was performed successfully (after that moment, the 3D element will move following the hand indicator). Since the element can float following the hand position it allows the user to move it in the workspace and add it to the program at a desired location.
- **Release:** The release process is performed in the opposite way and assumes that the system is in the grabbing state. It consists of an initial stage where the user' s palm hovers over the desired end location and a second stage where the user pushes the element towards the screen in 3D space to release it. Also, at that moment, the system is able to give the option to access features of the given element and "unlock" other actions once the process is finished, such as rotations or grab a new element.
- **Release incorrect item:** The process to release an incorrect grabbed item must be performed in the same way as the normal release, but in a different area of the screen, which will allow the user to detach the incorrectly selected item from the hand indicator and be able to pick a correct item.

5.3.2. Multithread interfacing framework analysis

The gesture definitions given above are used to 'construct' a program, but it is also necessary to clarify some specific elements related to the working environment of the proposed framework.

5.3.2.1. Metaphors analysis

Metaphors are essential to represent as much as possible, real components based on 3D elements. The hand gestures analysed previously and the graphic icons must facilitate the tasks and needs of a programmer instead of making it more difficult. This can be achieved by understanding the application area and automatically adapting the tools (Hurtienne et al., 2010). In our approach, the proposed framework's model developed is shown in the Figure 5.2. This figure represents a view of the main interface window, with the basic working elements.

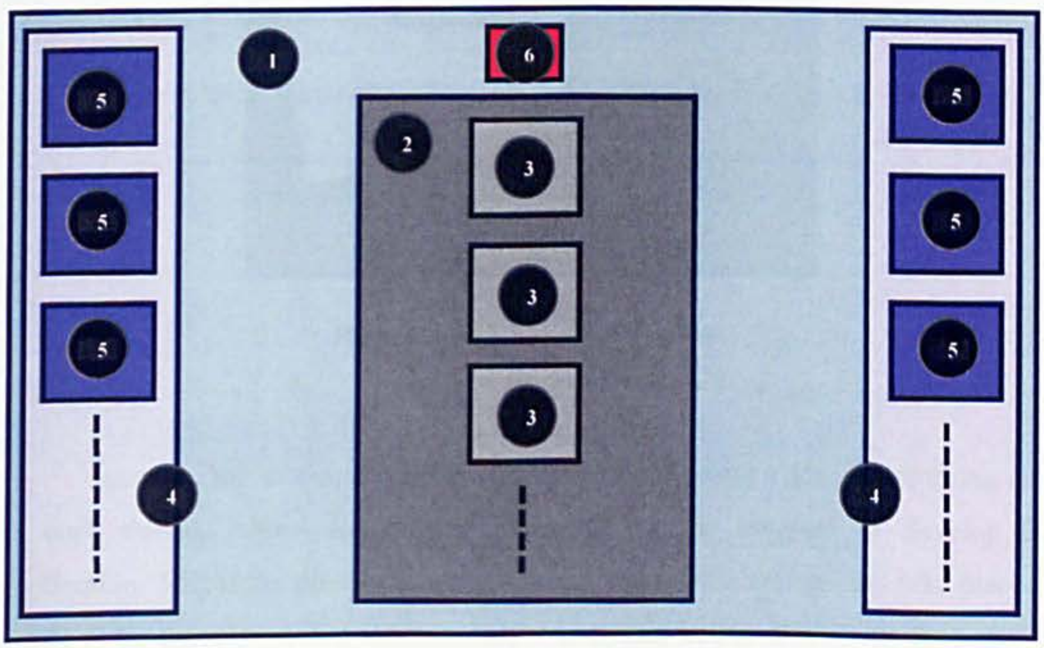


Figure 5.2: Programming graphic interface, where (1) corresponds to the interaction window; (2) corresponds to the programming area; (3) corresponds to the programming element's placement area; (4) corresponds to the area for the programming tools (5) corresponds to the specific space for each tool and (6) corresponds to the element release area.

As seen in Figure 5.2, the "programming area" is in the centre of the interface, with the programming elements (3D models that represent typical programming structures) on the left and right of the interface. Also, in the case of our experiments, an "example" area is added at the bottom left. Also, at the upper part of the working area, there is a "release element point", which allows the developer to release an undesired object that was previously selected. The elements and the interaction will be analysed in more detail in the following sections.

5.3.2.2. Main 3D Workspace

The main 3D workspace was presented in the previous section and corresponds to the area where the program will be created and this space is further divided according to the number of the supported threads. In the proposed interface prototype four threads are utilised, therefore the space is divided equally into four subareas as shown in Figure 5.3.

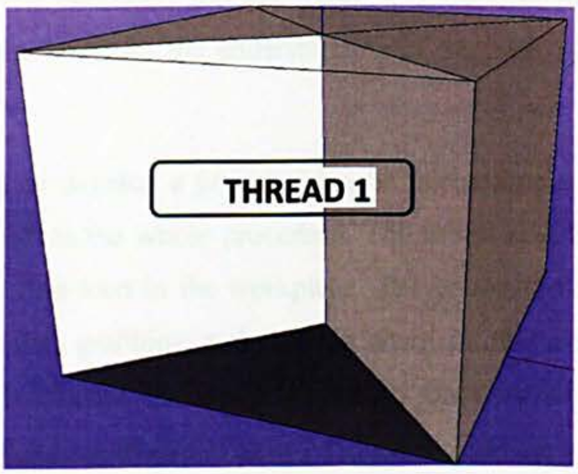


Figure 5.3: Workspace for four threads.

As seen, the workspace is divided equally providing a specific working area to each thread, where the graphic elements can be inserted to develop the application. This is the place where the rotation and release actions can take place. It needs to be mentioned that there is only one thread active at a given time and in order to move to the next one the rotation gesture needs to be performed. The previous configuration allows the development of software in parallel.

5.3.2.3. 3D iconic tools

The 3D iconic tools are the 3D models that represent basic programming elements such as conditions, variables, mathematical operators or other kind of functions.

These tools can be grabbed by the user to construct the program that will run in the selected thread.

5.3.3. Application development process

The presented 3D programming model aims to solve the problem related to the lack of visibility in the multi-thread programs, which limits the possibilities of the developer to see clearly how the threads connect with each other. Given the previous problem, a 3D iconic representation can be the solution, allowing the simultaneous visibility of different execution threads. Also, a 3D iconic metaphor, as it was discussed previously, increases the understanding of the problem and improves the suggested solutions.

The process to develop a program can be fairly simple and an example is provided to demonstrate the whole procedure. The initial step of the process is to grab and move the first icon in the workplace. The process of adding elements to each thread is similar, grabbing and moving them in the work area. After the completion of the first thread, the user just needs to rotate the workplace to continue with the following thread until all the tasks in all the threads are completed.

5.4. Analysis of the evaluation process

The evaluation process of the proposed framework is based on a simulated example that involves tasks related to the development of a specific multi-thread program. The proposed experiment provides information about two specific areas: performance and user satisfaction allowing both qualitative and quantitative evaluations. Additionally, a comparative study with the traditional development interfaces for multi thread applications is presented.

This experiment needs to provide mechanisms to evaluate all the features of the proposed framework and it is necessary to evaluate them using an example, simple enough to be understood by people who either do or do not have any experience in multi-thread programming. Based on these requirements, the task of adding an array of numbers using 4 threads was selected and the interface that was used is the one presented in the previous section. The C++ code for that task is presented in Figure 5.4. The experiment provides a set of tools necessary to program the code in each thread. The initialisation of values and variables is not part of this experiment.

```
//Global Variables
LL = 1000000
REPS = 1;
gloabal_sum=0.0;
aa[LL]
mutex_sum;

// Function that adds the elements of an array
void sum
{
    i= TID*(N/NT), i1= i0 + (N/NT);
    localSum=0.0;
    for (r=0; r<REPS; r++)
        for (i=i0; i<i1; i++)
            localSum += aa[i];

    mutex_sum.lock();
    global_sum = global_sum + localSum;
    mutex_sum.unlock();
}




main()
{
    // Initialization
    for (t=0; t<LL; t++)
    {
        aa[t] = t+1;
    }
    // Create 4 Threads
    thread_vector = new thread [4];
    NT = 4;
    for (t=0; t<NT; t++)
    {
        thread_vector[t] = thread(sum, t, LL, NT);
    }

    //Join the Threads
    for (t=0; t<NT; t++)
    {
        thrds[t].join;
    }

    // Display the result
    print global_sum
}
```

Figure 5.4: C++ code for multi-threaded framework's test.

The experiment follows the stages presented below:

- the ***presentation stage*** (where the interface is presented and explained to the users)
- the ***practice stage*** (where the users can interact with the interface and use the available features), and
- the ***task execution stage*** (where the users perform the task and quantitative factors are registered, such as task execution time and quantity of errors).

After that, the users are asked to complete a questionnaire with similar questions to the questionnaires provided by IBM's research about new interfaces (Lewis, 1993).

The main elements related to the experiment are analysed in the following sections.

5.4.2. Set of gestures

The set of gestures selected covers the interaction capabilities described in section 5.3.1. The set of gestures is defined as follows:

- **Rotation:** The gesture of rotation is to simply move the left hand from left to right. To avoid problems in understanding user' s interaction, the rotation is restricted to just one direction (from left to right) and it can be executed with just one arm. For our experiments, the left arm was selected.
- **Grab:** To grab elements, the user just needs to place the hand over an object and push. In that moment, the object will be "attached" to the hand indicator and move along with it. This gesture can be performed with both hands and allows grabbing two 3D objects simultaneously.
- **Release:** The process to release an element depends on the selected object. If the selected object corresponds to a desired element, it needs to be placed over the special contact points (programming slots). If the object is not the right one and it is not desired by the user, it can be released by placing it over the drop section and select another one.

For each of these actions, a set of thresholds is defined. These thresholds were selected experimentally, similar to the ones selected for the 3D database interaction mechanism proposed in chapter 4. Figure 5.5 helps to explain how the thresholds were defined.

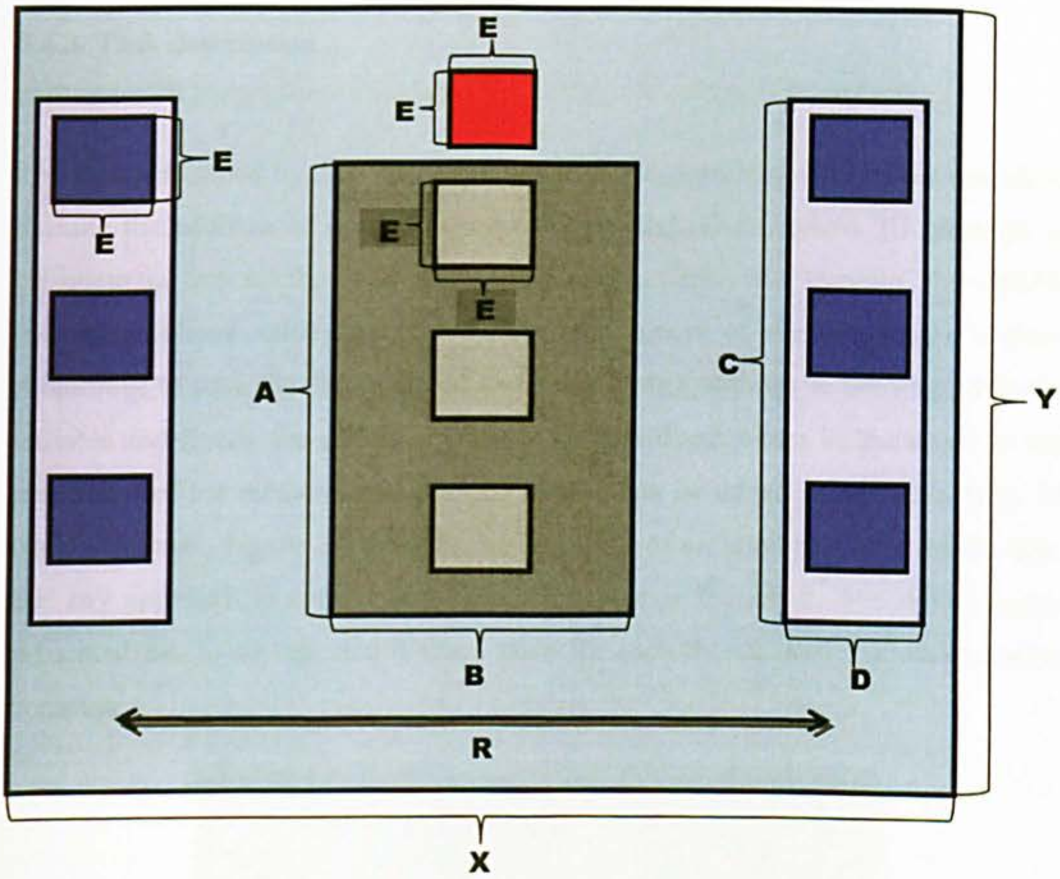


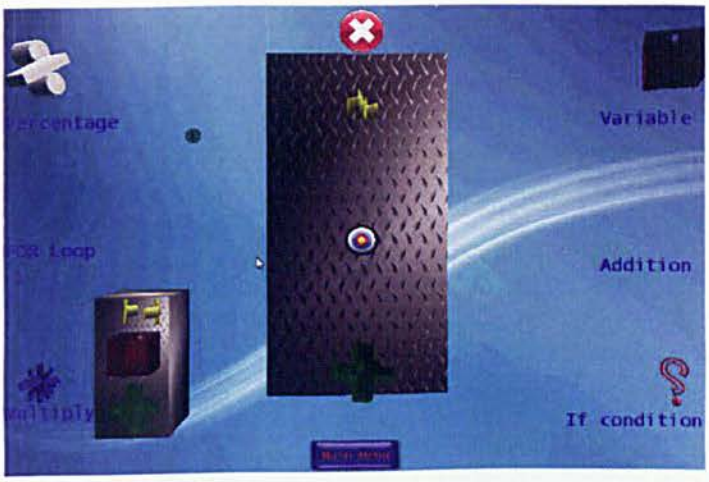
Figure 5.5: Interaction areas for the experimental interface, where the external light blue square represents the whole interaction area (with measures X×Y); the central rectangle represents the programming area (A×B) where each grey square represent the areas to place a programming 3D icon (E×E); the rectangles in the sides represent the toolboxes (C×D) where the blue squares represent the areas to grab the programming icon tools (E×E); the red square is the wrong element release area (E×E); and R represents the range of movement for rotation, that can be performed anywhere in interaction area.

Figure 5.5 shows a representation of interaction area on the screen (graphic interface) with height *Y* and width *X*; where the toolboxes have a height equal to *C* and width equal to *D* (using both approximately 30% of the interaction area); the programming area with height *A* and width *B* (using approximately the 24% of the interaction area) and the release area (red square on top) of width and height *E* (using the 3% of the interaction area). All the areas where the programming elements can be "grabbed from" or "placed in", are square with size *E*. *R* represents the length of the movement needed to rotate the programming area moving from one thread to the next one, which can be performed in any place of the interaction area.

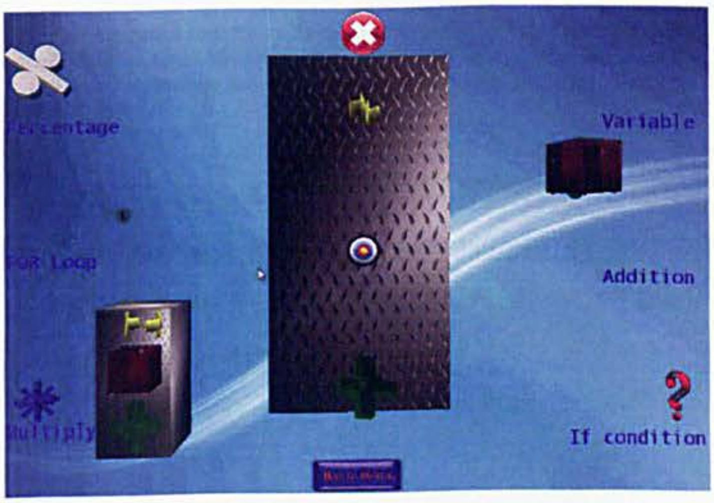
5.4.3. Task description

The task performed by the users consists of the programming all the four threads to execute the addition of several elements in parallel. The required 3D elements to complete the task are the 'loop cycle' graphic icon, which will "contain" the variable (i.e. accumulated value) and the summation process of the consecutive values. According to that, the first element to be selected (grabbing) is the loop, then the variable and finally the addition. There is no specific sequence in the selection and placement of the elements and two 3D objects can be added at the same time. To clarify the task, Figure 5.6 presents the sequence of an ideal execution of the task, but any approach is correct if the final outcome is the same. The same process presented has to be repeated 4 times once for each thread until the whole task is completed.

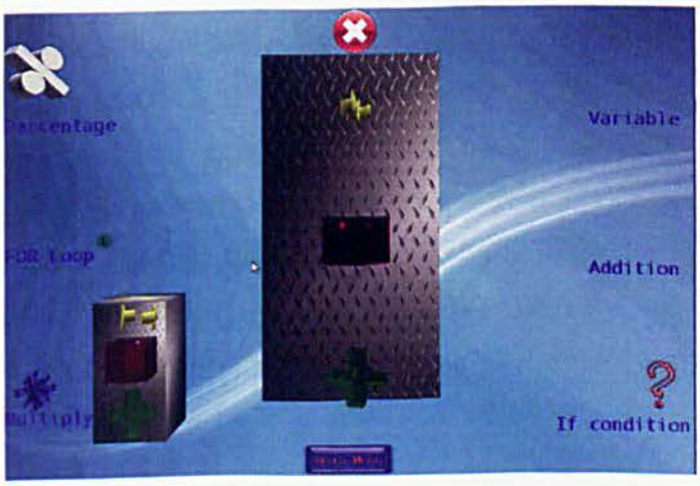
(a)


Percentage
Variable
FOR Loop
Addition
Multiply
If condition

(b)


Percentage
Variable
FOR Loop
Addition
Multiply
If condition

(c)


Percentage
Variable
FOR Loop
Addition
Multiply
If condition

(d)

(e)

Figure 5.6: Interaction sequence to create a thread in the interface a) first step: grab two elements. b) Place them in the work area c) Grab the third element d) Place the element in the corresponding position e) rotate the work area to start a new thread.

In this example the expected solution is provided to the users in the example box to help them with this task. At the end, the interface provides the required time to complete the whole process. In the following section, the experiment process and the evaluation procedure are presented.

5.4.3. Evaluation procedure

During the evaluation process the experiments were divided into several steps.

5.4.3.1 Present and explain the experiment and its objectives

The objective of this section is to provide information about the possibility of using 3D hand gesture interfaces instead of the traditional code based interaction to develop multi-threaded software. As a result, the proposed experiment performs a comparative study evaluating the users' performance on the proposed prototype interface versus the traditional approach. Since the users are informed about the overall concept of the experiment and its objectives, we are moving to the next step.

5.4.3.2 Demonstrate

The aim of the demonstration section is to present the interface and its elements to the users, answering any related questions. Also, if the users are not familiar with multi-thread programming, a short explanation and examples are given.

5.4.3.3 Familiarise the subject with the interface

During the familiarisation stage with the interface the interaction mechanism is presented to the users allowing them to practice with the basic movements and the on screen features.

5.4.3.4 Subject performs the available actions

The available functions (e.g. rotation, grabbing, etc.) are explained to the users and demonstrated in real time during this step. Furthermore, they are encouraged to practice and perform these functions by themselves. The total time of training is a few minutes, indicating also how much intuitive the proposed approach is.

5.4.3.5 Full task Performance

Since the users are familiar with the environment and with the mechanisms to perform the available functions, the full task that was initially introduced is performed counting the required time to successfully complete it, the number of errors during the users' interactions and the system errors due to erroneous action detection. The errors were considered according to the gestures described above. Therefore, the metrics to evaluate the interface were time to complete the task, number of user errors and number of system errors. Three types of errors where considered: grabbing, placement of the object and rotation of the work space.

5.4.3.6 Questionnaire completion

After the successful performance of the task, a questionnaire is completed by the users, evaluating and comparing the available interfaces (i.e. visual 3D and code based programming). The questionnaire was the tool to collect the users feedback and to provide a qualitative analysis. During this process any questions from the users are answered to make sure everything is clear to them, there is no time limit for this task.

The model of questionnaire that was used was based on the questionnaires provided by IBM in their research about new interfaces on usability tests.

The questionnaire is separated into two main sections in order to evaluate the user experience with the interface, shown in Table 5.1. In all the questions the users provide a number from 1 to 5 to evaluate the interface according to the question, where 1 is the lowest score (extremely negative evaluation) and 5 is the highest (extremely positive evaluation).

Table 5.1: Usability questionnaire for sections 1 and 2.

Section 1	Section 2 - How would you rate:
Q1: Was the interaction easy to understand?	Q4: The Interface?
Q2: Was it easy to manipulate?	Q5: The Performance?
Q3: Is the navigation system intuitive?	Q6: The functionality?
	Q7: The objective achieved?
	Q8: The user experience?
	Q9: The hand gestures selected?

At the end of the questionnaire, a last question is asked about the users' preference over the 3D visual gesture interface and the traditional code programming, where the users select their preferred approach (1 to the preferred interface and zero to the other). The obtained results are presented in the following section.

5.5. Experimental results

In order to evaluate the proposed interfaces, experiments were conducted using 29 subjects, aged 20 to 50 years of age. Regarding the subjects 63% were males and 37% were female. Also the level of programming knowledge and experience was well distributed among them from novice to experts. The result of the experiments will be separated into qualitative and quantitative analysis in both sections below. The results are shown in terms of medians and median absolute deviations to avoid the influence of outliers.

The statistical validation of the data obtained for the set of experiments conducted, was performed using the Wilcoxon signed/rank test. As happened with the experimental data in the previous chapters, our data cannot be considered normally distributed; given the level of skewness. However, in this case, given we have just a single experiment; the data obtained was compared with mean response values and a significance level of 0.01.

5.5.1. Qualitative results

In this section, the presented results correspond to the answers given by the users in the questionnaire previously discussed. The evaluation of the interface from the users is presented in the following tables and graphs. The Table 5.2 shows the median values for the questions answered, including the median absolute deviation over these values. In the first section, the best evaluated aspect by the users was the simplicity in understanding interaction with the interface. In general, the first section has high values, which indicates that the users found the interface intuitive and easy to use. For section 2, the evaluation is positive as well, with significantly high values for the interface, the achievement of the objective and the selected gestures, indicating the satisfaction of the users with the interface and the overall performance of the system.

Table 5.2: Median values and median absolute deviation in each question for all the interfaces.

<table>
<tr><th>S1</th><th colspan="2">Q1</th><th colspan="2">Q2</th><th colspan="2">Q3</th></tr>
<tr><td></td><td colspan="2">5.0 (0.0)</td><td colspan="2">4.0 (1.0)</td><td colspan="2">5.0 (0.0)</td></tr>
<tr><th>S2</th><th>Q4</th><th>Q5</th><th>Q6</th><th>Q7</th><th>Q8</th><th>Q9</th></tr>
<tr><td></td><td>5.0 (0.0)</td><td>4.0 (1.0)</td><td>4.0 (1.0)</td><td>5.0 (0.0)</td><td>5.0 (0.0)</td><td>5.0 (0.0)</td></tr>
</table>

Regarding the obtained values during the evaluation, all the aspects presented were ranked above 3.5, indicating the general acceptance of the new approaches based on hand gesture interaction.

Table 5.3 shows the results regarding the last qualitative question: the users' preference between the hand gesture interface and the traditional text-based interface to work with multiple-threads.

Table 5.3: Median values and median absolute deviation for final question (comparison between the 2 approaches).

	Gesture based	***Text based***
Median	1.0	0.0
Median absolute deviation	0.0	0.0

Most users generally prefer the hand gesture based approach over the text based interface because of the visual representation, which improves the understanding of the given task.

Regarding the qualitative analysis, the answers for the two sections versus different age ranges are shown in Figures 5.7 and 5.8. The median absolute deviation is added to the plots.

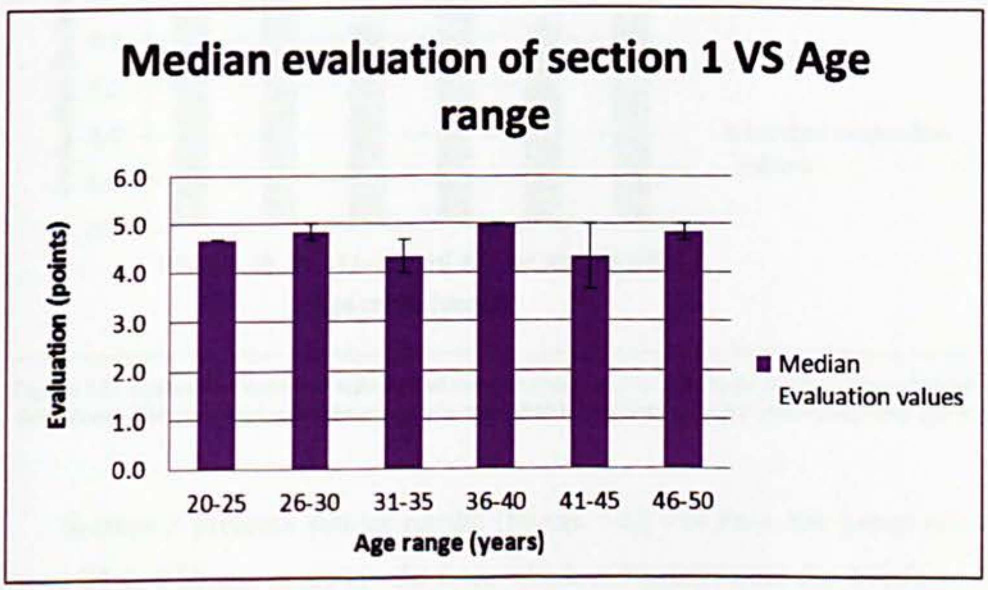


Figure 5.7: Evaluation values of section 1 of the questionnaire vs. age range (with median absolute deviation). The bars represent the scores given by different group of users (according their ages).

Figure 5.7, clearly indicates all subjects evaluated the interface itself with a value over 4 (very good) with a low median absolute deviation, showing that the evaluation given by the users is, in general, similar, which demonstrates a clear acceptance of the interface. The groups that evaluated the interface with the highest scores were the older users in the 36-40 age range, followed by users between 46 and 50, and users between 26 and 30 years of age. This can be explained by the intuitiveness of the interaction process that resembles other modern interaction interfaces (to the younger users) and provides an easier manipulation model (to the older users).

In this case, the statistical evaluation was performed with a mean response value of 3.5 (which indicates a positive evaluation of the interface). The values obtained were *Wilcoxon Statistic* = 6, $p < 0.01$, indicating a highly significant mean difference from the "neutral" response of 3.5 (one tailed), indicating the difference is highly positively significant for our approach.

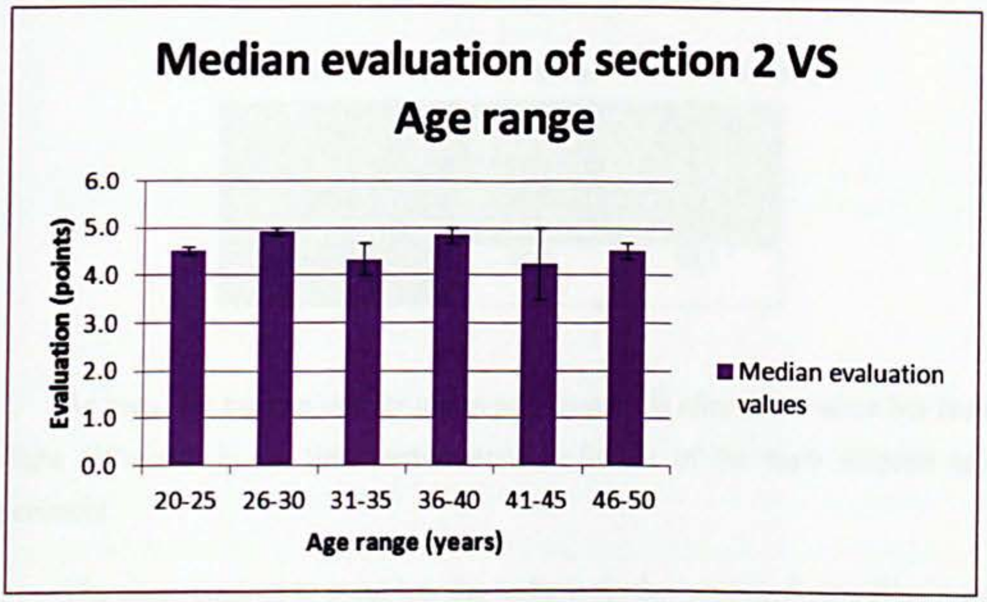


Figure 5.8: Evaluation values of section 2 of the questionnaire vs. age range (with median absolute deviation). The bars represent the scores given by different group of users (according their ages).

Section 2 presents similar results (Figure 5.8), but here, the group of users between 25 and 30 age range are the ones which evaluated better the interface. This result can be explained by the graphic definition of elements that are easier to understand and the distribution of the graphic elements, which provides the necessary space to interact properly.

The statistical evaluation was again performed with a mean response value of 3.5. In this case, *Wilcoxon Statistic* = 14, $p < 0.01$ (one tailed), indicating a highly significant mean difference from response of value, indicating the difference is highly positively significant for our approach.

5.5.2. Quantitative results

In this section the results obtained by measuring the completion time, the amount of user errors and the amount of errors from the interface are analysed.

5.5.2.1 Times

A summary of the average times to perform the task are shown in Table 5.4.

Table 5.4: Median time in seconds and age for males and females.

	Male	***Female***
Av. Age	32.0	32.5
Av. Time	90.0	92.7

As seen, the median age for males and females is almost the same but there is a slight difference in the time performance in favour of the male subjects in the experiment.

The time required to complete the tasks is further analysed providing a more detailed quantitative evaluation. The figures presented below show how different aspects are related to the speed and the time required to accomplish the tasks.

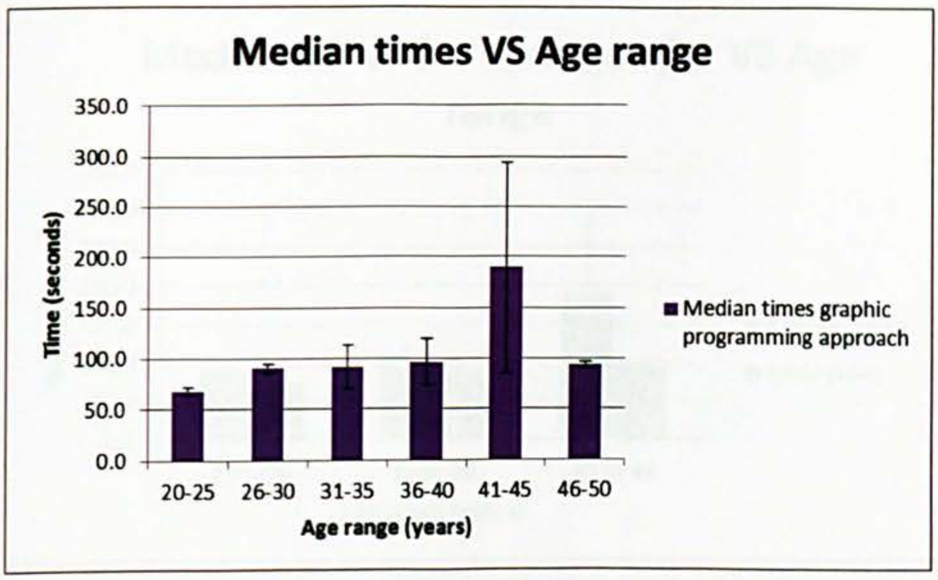


Figure 5.9: Median performance time of subjects based on their ages (with median absolute deviation). The bars represent the time to perform the task given different group of users (according their ages).

Figure 5.9, shows that the best performance corresponds to the subjects between 20 and 35 years old, with relatively low median absolute deviation. The subjects over that age present faster results (less time to perform the full task). The slowest performance was obtained by the group of users with ages between 41 and 45 years old; however, the median absolute deviation was really high for that group. That fact can be related to the speed of performance of the different gestures involved in the task, especially the grabbing and placement tasks.

The results of time performance presented by gender can be seen on Figure 5.10.

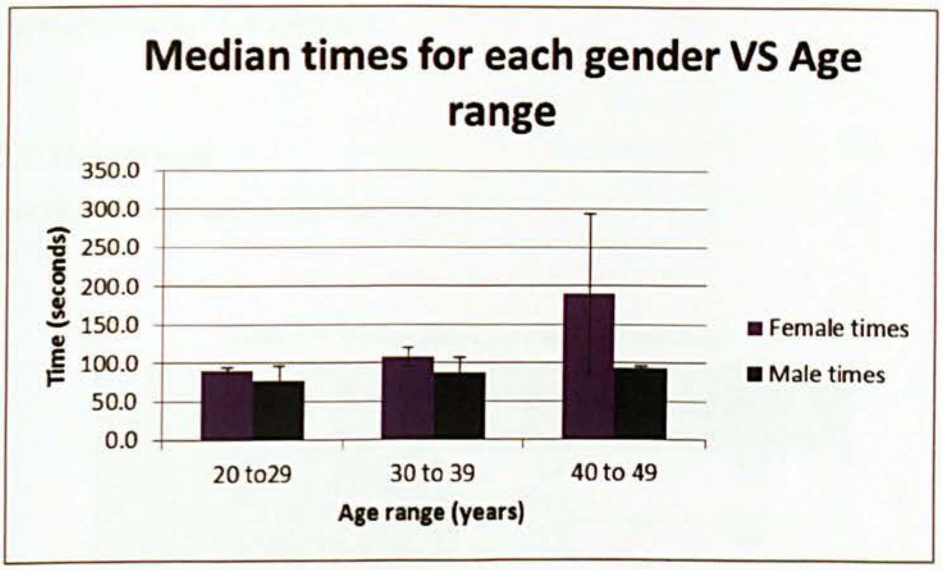


Figure 5.10: Performance time of male and female subjects based on their ages (with median absolute deviation). The blue bars represent the time to perform the task by male users and the purple bars represent the time to perform the task by female users, given different group of users (according their ages).

The female subjects (Figure 5.10) present their best performance (less than 20 seconds) in the range of 20 to 29 years of age, with a low deviation. The slowest performance was for the female subjects in the age range of 40 to 49, with times near 200 seconds, but with high median absolute deviation, indicating that the performance of the subjects in that age range varied probably because of the lack of understanding of the gestures or the speed to perform the grabbing and placing (as it was observed during the experiments, some users performed the displacement of the 3D icons really slowly compared with others). The case of male subjects is similar. As in the case of the female subjects, the male users with best times belong to the age range between 20 and 29 years old and the slowest results were obtained by the users between 40 and 49 years old. However, the times in the slowest case are better than in the female case and the median absolute deviation is a lot lower, indicating the performance of the users, especially on grabbing and placing the 3D icons, was significantly faster and consistent for all male users.

The statistical evaluation in this case was performed against the mean value of the time to perform the given task; 93.1 seconds. The values obtained were *Wilcoxon Statistic* = 134, $p > 0.01$ (two tailed), indicating that there is not statistical significant evidence to affirm the time results are more than the average value of the time performance for our approach.

5.5.2.2. *User Errors*

The user errors are summarized on Table 5.5.

Table 5.5: Median errors for males and females.

	Male	***Female***
Median Error Rotating	0.1	0.1
Median Error grabbing	0.5	1.0
Median Error Placing	0.2	0.0

The table above shows the average results for the 3 types of user errors: rotating the working area, grabbing the 3D programming elements and placing them in the correct locations. As it can be observed female subjects had better results than male subjects. Also, the highest value of error was in the process of grabbing an element. In Figure 5.11 the user errors over different age ranges are summarised.

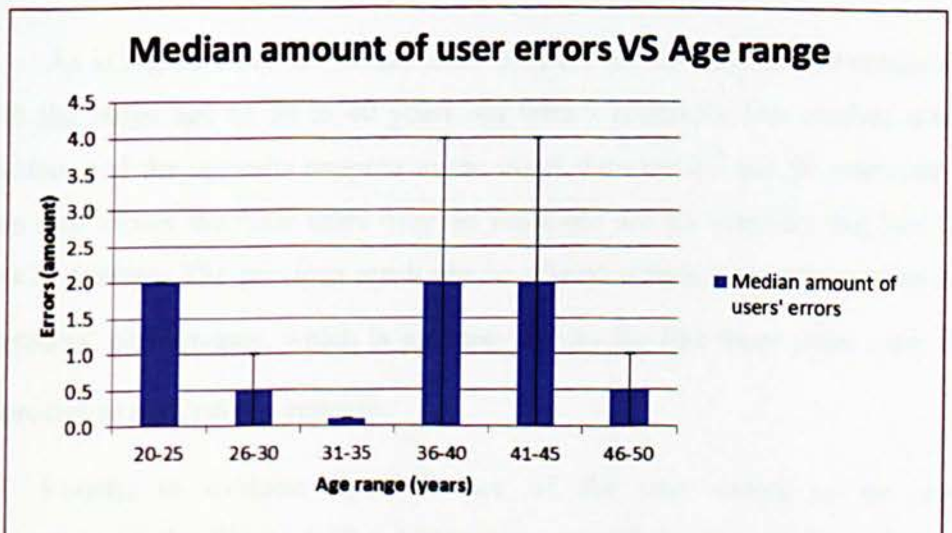


Figure 5.11: Median amount of user errors according the age range (with median absolute deviation). The blue bars represent the median amount of user errors, given different group of users (according their ages).

As it can be seen in Figure 5.11, the groups of users with the lowest number of errors are in the age range between 31 and 35 age range. Also, this group presents low median absolute deviation. This result can be related to the knowledge of similar programming tools.

In Figure 5.12 the median user errors for each gender are presented.

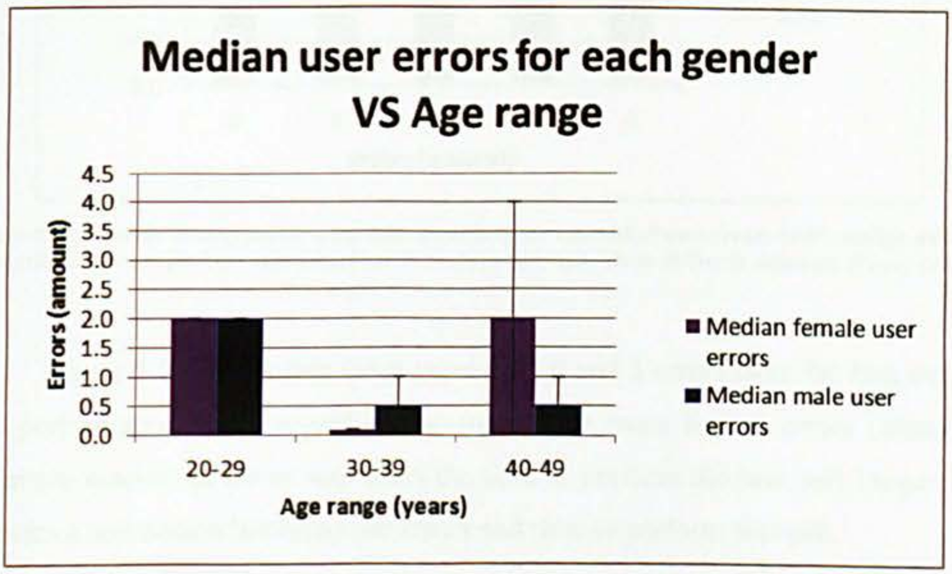


Figure 5.12: Median user errors according the age range by each gender (with median absolute deviation). The blue bars represent the amount of user errors for male subjects and the purple bars represent the amount of user errors for female subjects, given different group of users (according their ages).

As in Figure 5.12, the female users with the lowest number of average errors are in the range age of 30 to 40 years old with a relatively low median absolute deviation, and the opposite happens in the range between 40 and 50 years old. The figure also shows the male users over 40 years old are the subjects that had fewer errors in average. The previous result can be related with a more accurate sequence of gestures' performance, which is probably due to the fact these users were faster and precise to perform the gestures.

Finally, to evaluate the influence of the user errors in the overall performance, in the Figure 5.13 a comparison considering the number of errors is shown.

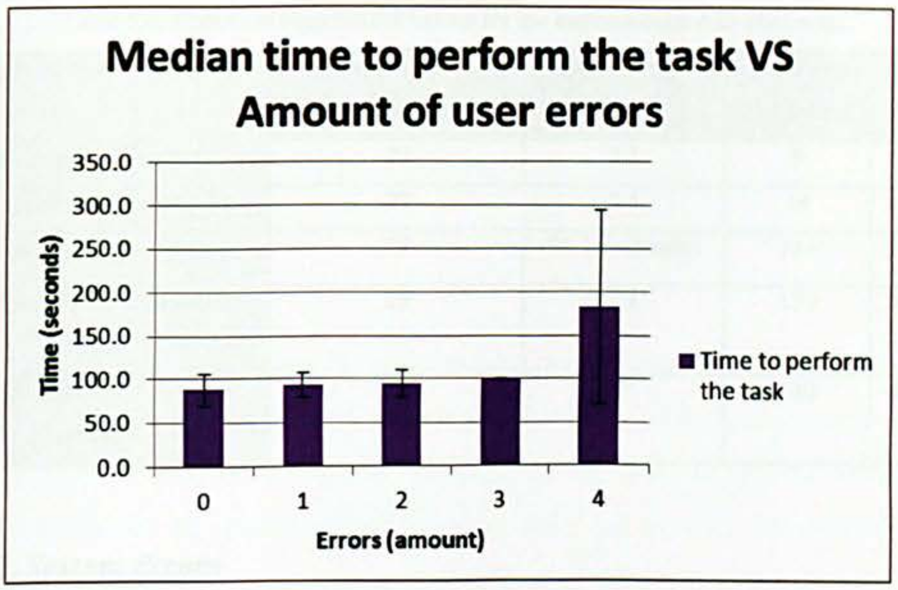


Figure 5.13: Median time to perform the task according the amount of user errors (with median absolute deviation). The purple bars represent time to perform the task, given different amounts of user errors.

Figure 5.13 shows that those users with 0 and 2 errors have the best average time performance. Also, regarding the users with more than 3 errors (where the maximum amount of errors was four) the time to perform the task was longer. This indicates a correlation between user errors and time to perform the task.

The statistical evaluation in this case was performed against the mean value of user errors, 1.1. The values obtained were *Wilcoxon Statistic* = 137, $p > 0.01$ (two tailed), indicating that there is not statistical significant evidence to affirm the amount of user errors are more than the average value of the time performance for our approach.

Finally, Table 5.6 shows the values obtained for our sets of experiments.

Table 5.6: Statistical significance values for the experimental data obtained.

Experimental Data	Amount of subjects involved (N)	Mean response value	Wilcoxon Statistical	P-Value
Questionnaire Section 1	29	3.5	6	0.00004
Questionnaire Section 2	29	3.5	14	0.00011
Average time performance	29	93.1 (seconds)	134	0.1161
Average amount of user errors	29	1.1	137	0.13286
Average amount of system errors	29	7	180	0.83818

5.5.2.3. System Errors

The final metric to be analysed corresponds to the errors generated by the system. These errors are associated to failure in identifying correctly a performed gesture. The results for this metric are shown in Figure 5.14.

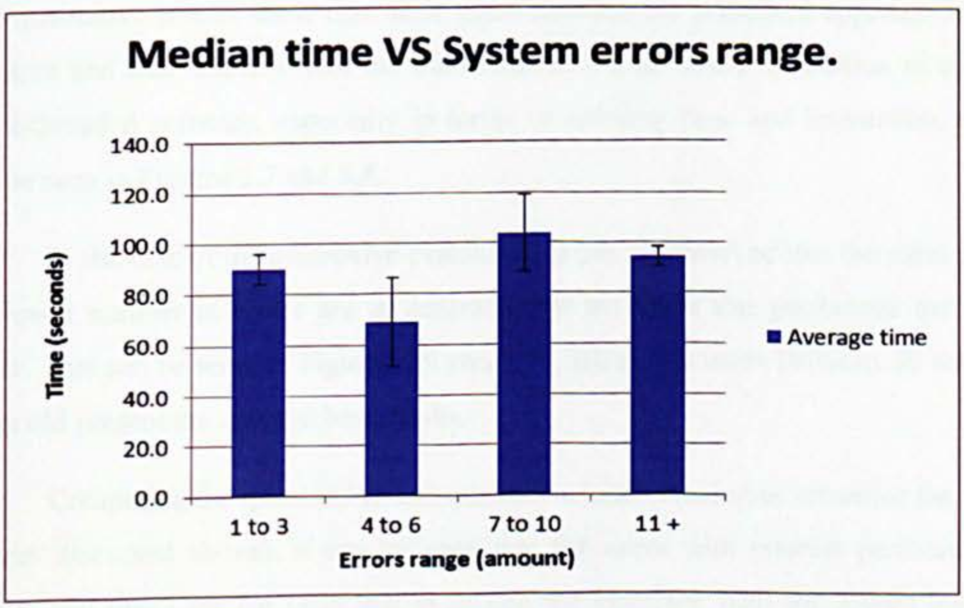


Figure 5.14: Median time according the amount of system errors clusters (with median absolute deviation). The blue bars represent time to perform the task, given different amounts of system errors.

Figure 5.14 presents the influence of the system errors in the overall required time to complete the tasks. As it can be seen, the amount of system errors in the range of 7 to 10 made the task more difficult, but with a high median absolute deviation, which indicates the average time performance in this case is not directly influenced by system errors. The error range between 4 and 6 has the best time performance. This indicates that the amount of system errors is not directly related to the time to perform the task, indicating that the users can relatively quickly overcome these kind of problems.

The statistical evaluation in this case was performed against the mean value of system errors, 7.In this case, the values obtained were *Wilcoxon Statistic* = 1802, $p > 0.01$ (two tailed), indicating that there is not statistical significant evidence to affirm the amount of system errors are more than the average value of the time performance for our approach.

5.5.3. Comparison of Qualitative and quantitative results

The qualitative results show that most users consider the presented approach more intuitive and user friendly than the traditional text code based interaction to create multi-threaded software, especially in terms of learning time and interaction, as it can be seen in Figures 5.7 and 5.8.

In the case of the qualitative evaluation, it can be observed that the users with the lower number of errors are in general terms the users that performed the task faster. This can be seen on Figures 5.9 and 5.11, where the users between 26 and 35 years old present the average best results.

Comparing the quantitative and qualitative results (using as reference the four figures discussed above), it can be seen that the users with poorest performance (errors and time) are the ones that evaluated the interface with the lowest marks, which corresponds to the users in the age range between 41 and 45 years old. In the next section, the conclusions of the chapter are presented.

5.6. Conclusions

In this chapter a novel 3D hand gesture based programming environment was presented, designed mainly for multi-thread applications. The complete definition of the interface involved graphic elements and advanced interaction techniques. The supported gestures and how to interact with the interface were analysed and a prototype graphic user interface was presented.

Also, in this chapter an evaluation experiment was described, analysing the evaluation procedure and metrics. The qualitative evaluation was based on a questionnaire to retrieve the users' opinions regarding the interface and the quantitative evaluation was based on three parameters: time to perform a specific task, number of user errors and number of system errors during the interaction.

The questionnaires analysis revealed that the users gave the system a positive evaluation, in particular, the system's performance and how much intuitive it was, confirming that a gesture based interface is more comfortable and easy to understand by the users.

According to the results obtained, it can be said that users of both genders have relatively similar performance on the performed execution times and amount of user errors. Also, the system errors do not affect significantly in the overall task execution time.

The results presented confirm that a gesture-based interface plus a 3D graphic development environment provide significant advantages in terms of performance and, consequently, can improve the user experience and the learning time of new programming techniques. Also, the 3D visualisation provides a better understanding of the task to design multi-thread software.

Our final chapter presents the conclusions and future work related with the research in 3D interaction presented, including a discussion of possible future works and challenges in the next generation interactive systems, aiming to create more natural interfaces based on 3D interaction.

Chapter 6

Conclusions and future work

In this thesis, novel frameworks to create and interact with 3D software using natural interfaces were proposed. The interaction models presented were based on the use of multi-touch devices and hand gestures. The main objective of this project was to generate a conceptual architecture for the development of 3D multi-touch and touchless hand gesture based interaction environments.

The main contributions of this research project are in the area of 3D human-computer interaction, focused on improving the techniques in 3D developing interfaces. The contributions addressed by this research project were:

- A novel framework to allow interaction between multiple portable multi-touch devices in 2D and 3D was proposed, considering multi-user collaborative interaction (chapter 3).
- Novel interaction styles for 3D data manipulation were defined, based on hand gestures trying to improve the user interfaces and to simplify the overall interaction procedures in 3D datasets (chapter 4).
- A novel framework for multi-thread programming on a 3D environment was suggested, improving the code efficiency, the overall robustness of the developed software and reducing the development and learning time (chapter 5).

A summary of the work presented in this thesis, the problems addressed and suggestions for future work are presented in the following sections.

6.1. Summary of work

In this section, an analysis of each chapter is presented, highlighting the achievements and conclusions on each of them.

In chapter 2 a review of HCI interfaces was presented, focusing on the next generation interaction approaches, such as multi-touch and hand gesture based systems, their graphic interfaces and the advantages they provide over traditional computer interaction mechanisms. The main aim of this chapter was to discuss the use of 3D interactions with 3D interfaces based on next generation interaction devices. These interfaces can improve several aspects in human-computer interaction, including reduction on training times for new interfaces, improvement in tasks' time performance, reduction of overall users' errors and improving the overall users' satisfaction, especially for data manipulation and programming environments. Also, the possibility of improving collaborative work was also discussed in this research work as an additional beneficial point of 3D interactive interfaces. The general conclusion in this chapter was the need to design and implement novel 3D interfaces with 3D interaction capabilities and demonstrate their superiority over traditional interaction mechanisms, collecting qualitative and quantitative feedback from users. The development of 3D interaction scenarios and collection of users' feedback were set as the main aims of the following chapters in this thesis.

In chapter 3, our aim was to prove the general advantages of 3D interfaces. To achieve that, a 3D movement-based interface to provide 3D interaction using portable touch devices was developed and evaluated against a similar interface based on traditional 2D touch interactions. To achieve the 3D movement interaction with a 3D interface, a set of experimental evaluations were performed to determine the possibility of using the inertial sensors from a mobile device. These experiments aimed to generate a 3D movement based interaction free of external elements, but the results proved that these devices are not accurate enough to provide the desired level of interaction. Given the negative results, it was necessary to generate a 3D interaction using a vision based system to achieve a proper 3D movement-based interface. A 3D interface was developed to test users' experience and performance given a simple task. The evaluation of this 3D interaction approach was done using two scenarios: single interaction and collaborative interaction, where the system was evaluated against a 2D touch based interface, which provided similar interaction capabilities as the 3D one. The experiment simulated a 3D sculpting process, where the users had to reveal a 3D shape by removing components from a 3D model. The aspects evaluated were both qualitative (using a questionnaire) and quantitative (considering time to perform the given task, amount of correct items removed and the number of incorrect items removed). For the single and collaborative cases, the 3D approach had better qualitative evaluation and provided faster times to complete the defined task and fewer incorrect actions than the 2D touch based interaction. In general, the 3D interaction approach proved its advantages over traditional 2D interaction in portable multi-touch devices, for single and collaborative interaction. Given the non-normally distributed nature of our data (since the obtained data curves present a high value of skewness), the Wilcoxon ranked signed test was applied to evaluate the statistical significance of our results. The Wilcoxon test was used in the following chapters as well.

The results obtained in chapter 3 showed the advantages provided by a 3D interaction method over simple 2D interaction. However, to provide a more realistic 3D interaction, the intermediate device (in our case, tablet PC and mobiles with touch capabilities) must be removed. The use of touchless hand gesture-based technologies can provide a 3D interaction without the use of external devices. Considering that, a novel 3D gesture based interface for 3D database interaction was presented, where the interaction with the system was performed using two hands-gesture interaction, providing the capability to perform 3D gestures. The evaluation of the framework was performed using two sets of gestures: one that considered 3D gestures and a second that was based only on 2D interactions. Also, these two set of gestures were compared with traditional SQL text-based code interaction for the same task. The evaluation for our experiments considered qualitative evaluation (provided by a questionnaire) and quantitative evaluation (using as metrics the time to perform the tasks, user errors to perform an interaction and system's errors when the users performed an interaction correctly). The experiments showed a clear preference for the hand gesture interaction over the traditional keyboard and mouse based interfaces. The 3D based approach related to the intuitive characteristics of the hand gesture interaction over a 3D environment received the best overall evaluation and the lowest performance times. Similarly, there is a correlation between the amount of users and system errors with the required time to perform the tasks. However, the number of errors in general is very low and does not affect significantly the users' performance and satisfaction, especially in the case of the 3D approach.

Also, the application of these interfaces in data manipulation systems indicate a better and faster understanding of the tasks by the users, mainly when the hand gesture interactions support 3D gestures. The 3D hand gesture interaction also simplifies the overall interaction process reducing number of steps to perform the tasks.

The experiments performed on chapter 3 and 4 allowed to illustrate the advantages of 3D interaction in given entertaining environments and data interaction systems respectively, based on 3D scenarios, concentrating on the application domain rather than on programming environments. Thus, chapter 5 presented a novel 3D hand gesture based programming environment designed for multi-thread applications and parallel computing. In this chapter, using the architecture defined in chapter 4, the complete definition of the interface involved only graphic elements and advanced interaction techniques were presented. The supported gestures and how to interact with the interface were analysed and a prototype graphic user interface was presented. The evaluation experiments consider the creation of a multi-threaded software, using a 3D environment, with 3D models to represent programming primitives. The interaction was based on hand gestures. The evaluation of the interface was qualitative (based on a questionnaire) and quantitative (based on time to perform the task, total number of user errors and total number of systems errors). The results presented confirm that a gesture-based interface plus a 3D graphic development environment provide significant advantages in terms of performance and, consequently, can improve the users' experience and the learning time of new developing mechanisms. Also, the overall users' evaluation of the interface was positive. The problems and advantages provided by the results of the experiments are discussed in the following section, regarding the 3D interaction.

6.2. Discussion

The creation of 3D users' interfaces generates several challenges, especially related to the scope of their use. The definition of the components and the interaction for the experimental evaluation in this research work considered the state-of-the-art devices for interaction, following current trends on HCI.

The use of multi-touch devices provided the starting point for analysing the possibilities of 3D interfaces based on 3D interactions, using vision-based algorithms implemented on portable devices (in our experiment, tablet PCs). The experiments proved that a 3D movement based interaction approach (especially in collaborative scenarios) can be successfully used. Furthermore, it provides a positive interaction experience to the users, which can be implemented in other application areas besides entertainment. The results can be improved changing the touch device for direct hand gesture interaction, generating more flexible environments.

Since there is not a large amount of research on touchless hand gestures-based interaction applied to data manipulation (as it was presented in the literature review), it was essential to perform experiments that provide information about 3D hand gesture manipulation and data interaction. This idea, plus the advances in new types of databases (such as graph databases or multidimensional databases) set the initial stage to generate a simple but effective interface based on hand gestures, allowing gesture interaction with a database, using a graphic 3D interaction scenario. The idea of utilising a cube to represent multiple sources of data appeared as a logic alternative, given the advances on multidimensional database interactive systems (as it was presented in section 4.3.3). The results indicated the advantages of the hand gesture interfaces in the presented scenario, especially in the case of the 3D hand gesture approach.

Regarding the results provided by previous sections and the advances on visual programming presented on the literature review, the set of experiments for a 3D hand gesture-based programming environment was developed, able to evaluate basic functionality for the creation of multi-threaded programming. The environment provided the tools necessary to prove our theories about the hand gesture interaction in a programming environment, obtaining useful information about the users' preference and the interaction mechanism. These experiments evaluated a simple interaction scenario; therefore the possibility of a more complex development environment can generate more interesting results. The result obtained of the experiment with this more complex environment can help to develop a new generation of programming environments.

In general terms, the work presented in this thesis achieved the objectives presented in the introduction (section 1.2) and provided experimental results to reinforce the hypotheses related to each experimental chapter.

Regarding the hand gesture interaction, new approaches to collect detailed information of hand postures are being developed. Better tracking and depth detection devices are being created too, which will provide new tools to improve the interaction environments presented in this thesis. The possibility of defining more natural hand gestures will lead to more realistic environments. The future work in the area will be presented in the following section.

6.3. Suggestions for future work

The advances in multi touch portable devices and more precise inertial sensors can be beneficial for future improvements on the collaborative environment presented, providing methods to define a 3D environment. This possibility will allow the use of the proposed interactive 3D mechanisms at any single and collaborative work environment where 3D interaction is suitable. Also, the evaluation process could include more users that will interact simultaneously, which will provide more useful data to analyse, addressing the advantages of collaborative interactions and their limits compared with the current interaction approaches in 3D touch environments.

In the case of 3D data manipulation, due to the modular nature of the presented architecture for hand gesture interfaces, the mechanisms for upgrading the presented interface are fairly simple, allowing the integration of future hand recognition technologies. The use of new devices and the definition of new sets of hand gestures with more detailed hand information (related to new advances on hand tracking and 3D modelling) can be added to the 3D interfaces presented to improve the overall results, reducing the system's errors during the detection. As was discussed, the use of more complex databases with more complex tasks could provide more information about how the 3D data representation on a natural interface can improve data manipulation and exploration tasks, and consequently, the user experience.

Similar achievements can be applied to our 3D programming approach. The generation of a 3D programming language using the definition presented in this thesis is also a possibility that could be explored in a future work. Providing definitions of more programming primitives, rules and the definition of structures that could be modified by the users to generate multi-threaded programming, can help to generate software in a more efficient way.

Maybe the most interesting area for future studies would be the combination of the use of touch and touchless hand gesture interaction. Since the reduction on the size of the devices capable of detecting depth, the combination of 3D hand gesture plus touch interaction when needed can be used in the creation of next generation interfaces that provide 3D interaction, data manipulation and a programming graphic 3D environment, similar to the project Real Sense presented by Intel and project Tango by Google, both this year.

The possibilities of development and research in this area aim to the creation of more complex integration of senses in the HCI. Due to the advance on interaction technology and algorithms for hand identification, new interaction systems will provide software platforms and interfaces, more suitable for human beings, based on the real world' s interaction. There will not be a need for large training sessions and the tasks will be achieved using metaphors of the real work, improving the creation of new and better software systems. All the previously presented alternatives can be part of future work related with this thesis.

Bibliography

- (Agrawal et al., 1997) Agrawal, R., Gupta, A., & Sarawagi, S. (1997, April). Modeling multidimensional databases. In Data Engineering, 1997. Proceedings. 13th International Conference on (pp. 232-243). IEEE.
- (Andreoli and De Chiara, 2005) Andreoli, R., De Chiara, R., Erra, U., & Scarano, V. (2005, July). Interactive 3d environments by using videogame engines. In Information Visualisation, 2005. Proceedings. Ninth International Conference on (pp. 515-520). IEEE.
- (Antle et al., 2011) Antle, A. N., Bevans, A., Tanenbaum, J., Seaborn, K., & Wang, S. (2011, January). Futura: design for collaborative learning and game play on a multi-touch digital tabletop. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (pp. 93-100). ACM.
- (Arth et al., 2011) Arth, C., Klopschitz, M., Reitmayr, G., & Schmalstieg, D. (2011, October). Real-time self-localization from panoramic images on mobile devices. In Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on (pp. 37-46). IEEE.

- (Azuma et al., 2001) Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. Computer Graphics and Applications, IEEE, 21(6), 34-47
- (Azuma, 1997) Azuma, R. T. (1997). A survey of augmented reality. Presence, 6(4), 355-385.
- (Baky, 2010) Baky, I. A. (2010). Solving multi-level multi-objective linear programming problems through fuzzy goal programming approach. Applied Mathematical Modelling, 34(9), 2377-2387.

- (Barnum, 2010) Barnum, C. M. (2010). Usability testing essentials: ready, set... test.

Elsevier.

- (Barr et al., 2007) Barr, P., Noble, J., & Biddle, R. (2007). Video game values: Human–computer interaction and games. Interacting with Computers, 19(2), 180-195.
- (Barret and Omote, 2010) Barrett, G., & Omote, R. (2010). Projected-capacitive touch technology. Information Display, 26(3), 16-21.
- (Bastien, 2010) Bastien, J. M. (2010). Usability testing: a review of some methodological and technical aspects of the method. International Journal of Medical Informatics, 79(4), e18-e23.
- (Benko et al., 2006) Benko, H., Wilson, A. D., & Baudisch, P. (2006, April). Precise selection techniques for multi-touch screens. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 1263-1272). ACM.
- (Benko et al., 2008) Benko, H., Wilson, A. D., & Balakrishnan, R. (2008, October). Sphere: multi-touch interactions on a spherical display. In Proceedings of the 21st annual ACM symposium on User interface software and technology (pp. 77-86). ACM.
- (Bier et al., 2013) Beier, F., Baumann, S., Betz, H., Hagedorn, S., & Wagner, T. (2013). Gesture-Based Navigation in Graph Databases–The Kevin Bacon Game. In BTW (pp. 511-514).
- (Biggs and MacDonald, 2003) Biggs, G., & MacDonald, B. (2003, December). A survey of robot programming systems. In Proceedings of the Australasian conference on robotics and automation (pp. 1-3).
- (Blackwell, 2000) Blackwell, A. F. (2000, December). Dealing with new cognitive dimensions. In Workshop on Cognitive Dimensions: Strengthening the Cognitive Dimensions Research Community, University of Hertfordshire.
- (Bohak and Marolt, 2013) Bohak, C., & Marolt, M. (2013). Kinect Web Kiosk Framework. In Human Factors in Computing and Informatics (pp. 785-790). Springer Berlin Heidelberg.
- (Böhm, 1998) Böhm, H. J. (1998). Prediction of binding constants of protein ligands: a fast method for the prioritization of hits obtained from de novo design or 3D database search programs. Journal of computer-aided molecular design, 12(4), 309-309.
- (Boring et al., 2011) Boring, S., Gehring, S., Wiethoff, A., Blöckner, A. M., Schöning, J., & Butz, A. (2011, May). Multi-user interaction on media facades

through live video on mobile devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2721-2724). ACM.

- (Brown, 1988) Brown, C. M. (1999). Human-computer interface design guidelines. Intellect Books.
- (Burnett et al., 1995) Burnett, M. M., Baker, M. J., Bohus, C., Carlson, P., Yang, S., & Van Zee, P. (1995). Scaling up visual programming languages. Computer, 28(3), 45-54.
- (Buxton et al., 1985) Buxton, W., Hill, R., & Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. ACM SIGGRAPH Computer Graphics, 19(3), 215-224.
- (Buxton, 2010) Buxton, B. (2010, May). 31.1: Invited Paper: A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future. In SID Symposium Digest of Technical Papers (Vol. 41, No. 1, pp. 444-448). Blackwell Publishing Ltd.
- (Calentano and Minuto, 2008) Celentano, A., & Minuto, A. (2008, September). Gestures, Shapes and Multitouch Interaction. In Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on (pp. 137-141). IEEE.
- (Canlion and Billard, 2007) Calinon, S., & Billard, A. (2007, March). Incremental learning of gestures by imitation in a humanoid robot. In Proceedings of the ACM/IEEE international conference on Human-robot interaction (pp. 255-262). ACM.
- (Carlin et al., 1997) Carlin, A. S., Hoffman, H. G., & Weghorst, S. (1997). Virtual reality and tactile augmentation in the treatment of spider phobia: a case report. Behaviour research and therapy, 35(2), 153-158.
- (Carmigniani and Furht, 2011) Carmigniani, J., & Furht, B. (2011). Augmented reality: an overview. In Handbook of augmented reality (pp. 3-46). Springer New York.

- (Chang et al., 2010) Chang, R., Wang, F., & You, P. (2010, April). A Survey on the Development of Multi-touch Technology. In Wearable Computing Systems (APWCS), 2010 Asia-Pacific Conference on (pp. 363-366). IEEE.
- (Chau et al., 2013) Chau, M., Wong, A., Wang, M., Lai, S., Chan, K. W., Li, T. M., ... & Sung, W. K. (2013). Using 3D virtual environments to facilitate students in constructivist learning. Decision Support Systems, 56, 115-121.

- (Chaudhary et al., 2013) Chaudhary, A., Raheja, J. L., Das, K., & Raheja, S. (2013). Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey. arXiv preprint arXiv:1303.2292.

- (Chaudhuri and Dayal, 1997) Chaudhuri, S., & Dayal, U. (1997). An overview of data warehousing and OLAP technology. ACM Sigmod record, 26(1), 65-74.
- (Chen et al., 2000) Chen, D. J., Koong, C. S., Chen, W. C., Huang, S. K., & Van Diepen, N. W. P. (2000). Integration of reusable software components and frameworks into a visual software construction approach. J. Inf. Sci. Eng., 16(6), 863-884.
- (Chen et al., 2012) Cheng, L., Sun, Q., Su, H., Cong, Y., & Zhao, S. (2012, May). Design and implementation of human-robot interactive demonstration system based on Kinect. In Control and Decision Conference (CCDC), 2012 24th Chinese (pp. 971-975). IEEE.
- (Chen, 2006) Chen, C. (2006). Virtual Environments. Information Visualization: Beyond the Horizon, 211-254.
- (Cheng et al., 2009) Cheng, I., Michel, D., Argyros, A., & Basu, A. (2009, October). A HIMI model for collaborative multi-touch multimedia education. In Proceedings of the 2009 workshop on Ambient media computing (pp. 3-12). ACM.
- (Chittaro and Ranon, 2007) Chittaro, L., & Ranon, R. (2007). Web3D technologies in learning, education and training: Motivations, issues, opportunities. Computers & Education, 49(1), 3-18.
- (Chittaro et al, 2009) Chittaro, L., Ranon, R., & Ieronutti, L. (2009, November). 3D object arrangement for novice users: the effectiveness of combining a first-person and a map view. In Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (pp. 171-178). ACM.
- (Chow et al., 2011)Chow, E., Hammad, A., & Gauthier, P. (2011, May). Multi-touch screens for navigating 3D virtual environments in participatory urban planning. In CHI'11 Extended Abstracts on Human Factors in Computing Systems (pp. 2395-2400). AC
- (Clerici et al., 2009) Clerici, S., Zoltan, C., & Prestigiacomo, G. (2009). Nimotoons: a totally graphic workbench for program tuning and experimentation. Electronic Notes in Theoretical Computer Science, 258(1), 93-107.
- (Clifton et al., 2010) Clifton, P., Mazalek, A., LCC, G. T., Nitsche, M., Sanford, J., COA, G. T., & Murray, J. (2010). WorkTop: A Multi-touch Tabletop Collaborative Sketching Application for Interdisciplinary Design (Doctoral dissertation, Master Thesis Georgia Institute of Technology).

- (Cockburn, 2004) Cockburn, A. (2004, January). Revisiting 2D vs 3D implications on spatial memory. In Proceedings of the fifth conference on Australasian user interface-Volume 28 (pp. 25-31). Australian Computer Society, Inc.
- (Coelho and Verbeek, 2014) Coelho, J. C., & Verbeek, F. J. (2014). Pointing Task Evaluation of Leap Motion Controller in 3D Virtual Environment. Creating the Difference, 78.
- (Conway et al., 2000) Conway, M., Audia, S., Burnette, T., Cosgrove, D. and Christiansen, K. (2000) 'Alice: lessons learned from building a 3D system for novices', CHI '00 Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 486-493.
- (Corke, 1993) Corke, P. I. (1993). Visual control of robot manipulators-a review. Visual servoing, 7, 1-31.
- (Crivelli et al., 2004) Crivelli, S., Kreylos, O., Hamann, B., Max, N., & Bethel, W. (2004). ProteinShop: a tool for interactive protein manipulation and steering. Journal of computer-aided molecular design, 18(4), 271-285.
- (Dang et al., 2009) Dang, N. T., Tavanti, M., Rankin, I., & Cooper, M. (2009). A comparison of different input devices for a 3D environment. International Journal of Industrial Ergonomics, 39(3), 554-563.
- (Davidson and Han, 2006) Davidson, P. L., & Han, J. Y. (2006, June). Synthesis and control on large scale multi-touch sensing displays. In Proceedings of the 2006 conference on New interfaces for musical expression (pp. 216-219). IRCAM—Centre Pompidou.
- (de la Riviere et al., 2008) de la Rivière, J. B., Kervégant, C., Orvain, E., & Dittlo, N. (2008, October). CubTile: a multi-touch cubic interface. In Proceedings of the 2008 ACM symposium on Virtual reality software and technology (pp. 69-72). ACM.
- (Della Penna et al., 2013) Della Penna, G., Magazzeni, D., & Orefice, S. (2013). A general theory of spatial relations to support a graphical tool for visual information extraction. Journal of Visual Languages & Computing, 24(2), 71-87.
- (Dhawale et al., 2006) Dhawale, P., Masoodian, M., & Rogers, B. (2006, July). Bare-hand 3D gesture input to interactive systems. In Proceedings of the 7th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: design centred HCI (pp. 25-32). ACM.
- (Di Pietro et al., 2008) Dipietro, L., Sabatini, A. M., & Dario, P. (2008). A survey of glove-based systems and their applications. Systems, Man, and Cybernetics, Part C:

Applications and Reviews, IEEE Transactions on, 38(4), 461-482.

- (Dietz and Leigh, 2001) Dietz, P., & Leigh, D. (2001, November). DiamondTouch: a multi-user touch technology. In Proceedings of the 14th annual ACM symposium on User interface software and technology (pp. 219-226). ACM.
- (Dillenbourg and Evans, 2011)Dillenbourg, P., & Evans, M. (2011). Interactive tabletops in education. International Journal of Computer-Supported Collaborative Learning, 6(4), 491-514.
- (Dix, 2009) Dix, A. (2009). Human-computer interaction (pp. 1327-1331). Springer US.
- (Dixon et al., 2006) Dixon, S. L., Smondyrev, A. M., Knoll, E. H., Rao, S. N., Shaw, D. E., & Friesner, R. A. (2006). PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. Journal of computer-aided molecular design, 20(10-11), 647-671.
- (Dohse et al., 2008) Dohse, K. C., Dohse, T., Still, J. D., & Parkhurst, D. J. (2008, February). Enhancing multi-user interaction with multi-touch table-top displays using hand tracking. In Advances in Computer-Human Interaction, 2008 First International Conference on (pp. 297-302). IEEE.
- (Dolbeau et al., 2007) Dolbeau, R., Bihan, S., & Bodin, F. (2007, October). HMPP: A hybrid multi-core parallel programming environment. In Workshop on General Purpose Processing on Graphics Processing Units (GPGPU 2007).

- (Döring et al., 2010) Döring, T., Shirazi, A. S., & Schmidt, A. (2010, May). Exploring gesture-based interaction techniques in multi-display environments with mobile phones and a multi-touch table. In AVI (Vol. 10, pp.419-419).
- (Echtler and Klinker, 2008) Echtler, F., & Klinker, G. (2008, October). A multi touch software architecture. In Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges (pp. 463-466). ACM.

- (Edmondson and Beale, 2008) Edmondson, W. H., & Beale, R. (2008). Projected Cognition–extending Distributed Cognition for the study of human interaction with computers. Interacting with Computers, 20(1), 128-140.
- (Epps et al., 2006) Epps, J., Lichman, S., & Wu, M. (2006, April). A study of hand shape use in tabletop gesture interaction. In CHI'06 extended abstracts on human factors in computing systems (pp. 748-753). ACM.
- (Erol et al., 2007) Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., & Twombly, X. (2007). Vision-based hand pose estimation: A review. Computer Vision and Image

Understanding, 108(1), 52-73.

- (Escudeiro et al., 2013) Escudeiro, P., Escudeiro, N., Reis, R., Barbosa, M., Bidarra, J., Baltazar, A. B., & Gouveia, B. (2013). Virtual Sign Translator.
- (Esnault et al., 2010) Esnault, N., Royan, J., Cozot, R., & Bouville, C. (2010, July). A flexible framework to personalize 3d web users experience. In Proceedings of the 15th International Conference on Web 3D Technology (pp. 35-44). ACM.
- (Fernquist et al., 2010) Fernquist, J., Booth, K. S., Mackworth, A. K., Kellett, R., & Girling, C.(2010, May). Using Multi-touch Tabletops to Create and Compare Neighbourhood Designs that Satisfy Constraints. In 2nd International Conference on Computational Sustainability, Cambridge, MA.
- (Fiorella et al., 2010) Fiorella, D., Sanna, A., & Lamberti, F. (2010). Multi-touch user interface evaluation for 3D object manipulation on mobile devices. Journal on Multimodal User Interfaces, 4(1), 3-10.
- (Fishkin, 2004) Fishkin, K. P. (2004). A taxonomy for and analysis of tangible interfaces. Personal and Ubiquitous Computing, 8(5), 347-358
- (Francese et al., 2012) Francese, R., Passero, I., & Tortora, G. (2012, May). Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI. In Proceedings of the International Working Conference on Advanced Visual Interfaces (pp. 116-123). ACM.
- (Fratti and Prattichizzo, 2011) Frati, V., & Prattichizzo, D. (2011, June). Using Kinect for hand tracking and rendering in wearable haptics. In World Haptics Conference (WHC), 2011 IEEE (pp. 317-321). IEEE.
- (Geiger et al, 1998) Geiger, C., Mueller, W., & Rosenbach, W. (1998, September). SAM-an animated 3D programming language. In Visual Languages, 1998. Proceedings. 1998 IEEE Symposium on (pp. 228-235). IEEE.
- (George et al., 2011) George, J., de Araujo, E., Dorsey, D., McCrickard, D. S., & Wilson, G. (2011). Multi touch tables for collaborative object-based learning. In Design, User Experience, and Usability. Theory, Methods, Tools and Practice (pp. 237-246). Springer Berlin Heidelberg.
- (Gill and Tomar, 2010) Gill, N. S., & Tomar, P. (2010). Modified development process of component-based software engineering. ACM SIGSOFT Software Engineering Notes, 35(2), 1-6.
- (Goldberg et al., 1999) Goldberg, J. H., & Kotval, X. P. (1999). Computer interface evaluation using eye movements: methods and constructs. International Journal of Industrial Ergonomics, 24(6), 631-645.

- (Gomez et al., 2012) Gomez, L. I., Gómez, S. A., & Vaisman, A. A. (2012, March). A generic data model and query language for spatiotemporal olap cube analysis. In Proceedings of the 15th International Conference on Extending Database Technology (pp. 300-311). ACM.
- (Goodman, 1999) Goodman, SN (1999). "Toward Evidence-Based Medical Statistics. 2: The Bayes factor..". Annals of Internal Medicine 130: 1005–1013.
- (Green et al., 1996) Green, T. R. G., & Petre, M. (1996). Usability analysis of visual programming environments: a 'cognitive dimensions' framework. Journal of Visual Languages & Computing, 7(2), 131-174.
- (Grigoreanu and Mohanna, 2013) Grigoreanu, V., & Mohanna, M. (2013, April). Informal cognitive walkthroughs (ICW): paring down and pairing up for an agile world. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 3093-3096). ACM.
- (Gu et al., 2010) Gu, N., Kim, M. J., & Maher, M. L. (2011). Technological advancements in synchronous collaboration: The effect of 3D virtual worlds and tangible user interfaces on architectural design. Automation in Construction, 20(3), 270-278.
- (Guettard et al., 2011) Guettard, E., Jandziak, S., Lombard, C., Podevin, N., Varane, V., & Dumont, P. (2011). Virtual reality from the keyboard/mouse couple to Kinect. Annals of Physical and Rehabilitation Medicine, 54(10 suppl. 1), 239.
- (Han, 2005) Han, J. Y. (2005, October). Low-cost multi-touch sensing through frustrated total internal reflection. In Proceedings of the 18th annual ACM symposium on User interface software and technology (pp. 115-118). ACM.
- (Han, 2006) Han, J. Y. (2006, July). Multi-touch interaction wall. In ACM SIGGRAPH 2006 Emerging technologies (p. 25). ACM.
- (Hansen and Hourcade, 2010) Hansen, T. E., & Hourcade, J. P. (2010, November). Comparing multi-touch table-tops and multi-mouse single-display groupware setups. In Proceedings of the 3rd Mexican Workshop on Human Computer Interaction (pp. 36-43). Universidad Politécnica de San Luis Potosí.
- (Hansen et al., 2009) Hansen, T. E., Hourcade, J. P., Virbel, M., Patali, S., & Serra, T. (2009, November). PyMT: a post-WIMP multi-touch user interface toolkit. In Proceedings of the ACM International Conference on Interactive Table-tops and Surfaces (pp. 17-24). ACM.
- (Hardy and Rukzio, 2008) Hardy, R., & Rukzio, E. (2008, September). Touch & interact: touch-based interaction of mobile phones with displays. In Proceedings of the 10th international conference on Human computer interaction with mobile

devices and services (pp. 245-254). ACM.

- (Harris et al., 1998) Harris, M., Buxton, B., Freeman, W. T., Ishii, H., Lucente, M., & Sinclair, M. J. (1998, July). Interfaces for humans (panel): natural interaction, tangible data, and beyond. In ACM SIGGRAPH 98 Conference abstracts and applications (pp. 200-202). ACM.
- (Harris et al., 2009) Harris, A., Rick, J., Bonnett, V., Yuill, N., Fleck, R., Marshall, P., & Rogers, Y. (2009, June). Around the table: are multiple-touch surfaces better than single-touch for children's collaborative interactions?. In Proceedings of the 9th international conference on Computer supported collaborative learning-Volume 1 (pp. 335-344). International Society of the Learning Sciences.
- (Harrow, 2000) Harrow Jr., J. J. (2000). Runtime checking of multithreaded applications with visual threads. In SPIN Model Checking and Software Verification (pp. 331-342). Springer Berlin Heidelberg.
- (Hernadez-Lopez et al., 2012) Hernández-López, J. J., Quintanilla-Olvera, A. L., López-Ramírez, J. L., Rangel-Butanda, F. J., Ibarra-Manzano, M. A., & Almanza-Ojeda, D. L. (2012). Detecting objects using color and depth segmentation with Kinect sensor. Procedia Technology, 3, 196-204.
- (Hersovic et al., 2011) Herskovic, V., Ochoa, S. F., Pino, J. A., & Neyem, H. A. (2011). The Iceberg Effect: Behind the User Interface of Mobile Collaborative Systems. J. UCS, 17(2), 183-201.
- (Heun et al., 2013) Heun, V., Kasahara, S., & Maes, P. (2013, April). Smarter objects: using AR technology to program physical objects and their interactions. In CHI'13 Extended Abstracts on Human Factors in Computing Systems (pp. 961-966). ACM.
- (Higgins et al, 2012) Higgins, S., Mercier, E., Burd, L., & Joyce-Gibbons, A. (2012). Multi-touch tables and collaborative learning. British Journal of Educational Technology, 43(6), 1041-1054.
- (Higgins et al., 2011) Higgins, S. E., Mercier, E., Burd, E., & Hatch, A. (2011). Multi-touch tables and the relationship with collaborative classroom pedagogies: A synthetic review. International Journal of Computer-Supported Collaborative Learning, 6(4), 515-538.
- (Hirte et al, 2012) Hirte, S., Seifert, A., Baumann, S., Klan, D., & Sattler, K. U. (2012, April). Data3--a Kinect interface for OLAP using complex event processing. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on (pp. 1297-1300). IEEE.

- (Hornecker et al., 2008) Hornecker, E., Marshall, P., Dalton, N. S., & Rogers, Y. (2008, November). Collaboration and interference: awareness with mice or touch input. In Proceedings of the 2008 ACM conference on Computer supported cooperative work (pp. 167-176).ACM.
- (Hsin et al., 2011) Hsin, K. Y., Morgan, H. P., Shave, S. R., Hinton, A. C., Taylor, P., & Walkinshaw, M. D. (2011). EDULISS: a small-molecule database with data-mining and pharmacophore searching capabilities. Nucleic acids research, 39(suppl 1), D1042-D1048.
- (Huang et al., 2011) Huang, T. S., Hua, G., Yang, M. H., Learned-Miller, E., Ma, Y., Turk, M., & Kriegman, D. J. (2011). Introduction to the special section on real-world face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(10), 1921-1924.
- (Hurtienne et al., 2010) Hurtienne, J., Stößel, C., Sturm, C., Maus, A., Rötting, M., Langdon, P., & Clarkson, J. (2010). Physical gestures for abstract concepts: Inclusive design with primary metaphors. Interacting with Computers, 22(6), 475-484.
- (Idreos and Liarou, 2013) Idreos, S., & Liarou, E. (2013). dbTouch: Analytics at your Fingertips. In CIDR.
- (Irawati et al., 2005) Irawati, S., Calderón, D., & Ko, H. (2005, December). Semantic 3D object manipulation using object ontology in multimodal interaction framework. In Proceedings of the 2005 international conference on Augmented tele-existence (pp. 35-39). ACM.
- (Ishii and Ullmer, 1997) Ishii, H., & Ullmer, B. (1997, March). Tangible bits: towards seamless interfaces between people, bits and atoms. In Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (pp. 234-241). ACM.
- (Ishii, 2008) Ishii, H. (2008) Tangible User Interface and Its Evolution PDF PDF.
- (Itoh and Tanaka, 2006) Itoh, M., & Tanaka, Y. (2006, December). 3d component-based visualization framework for generating simple 3d applications using web services. In Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on (pp. 823-830). IEEE.
- (Jackson et al., 2012) Jackson, B., Schroeder, D., & Keefe, D. F. (2012, May). Nailing down multi-touch: anchored above the surface interaction for 3D modelling and navigation. In Proceedings of Graphics Interface 2012 (pp. 181-184). Canadian Information Processing Society.

- (Jaday and Panchal, 2012) Jadav, J. J., & Panchal, M. (2012). Association rule mining method on OLAP cube. International Journal of Engineering Research and Applications (IJERA), 2(2), 1147-1151.

- (Jaimes and Sebe, 2007) Jaimes, A., Sebe, N. (2007). Multimodal human–computer interaction: A survey. Computer vision and image understanding, 108(1), 116-134.
- (Jaimes et al., 2007) Jaimes, A., Gatica-Perez, D., Sebe, N., Huang, T. S. (2007). Human-centered computing: toward a human revolution. IEEE Computer, 40(5), 30-34.
- (Jang et al., 2007) Jang, H. B., Kim, J. W., & Lee, C. W. (2007). Augmented reality cooking system using table-top display interface. In International Symposium on Ubiquitous VR (p. 1).
- (Jeffries et al, 1991) Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. (1991, March). User interface evaluation in the real world: a comparison of four techniques. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 119-124). ACM.
- (Jia et al., 2009) Jia, D., Bhatti, A., & Nahavandi, S. (2009, November). MUSTe method for quantifying virtual environment training system efficacy. In Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (pp. 305-308). ACM.
- (Jia et al., 2013) Jia, D., Bhatti, A., Nahavandi, S., & Horan, B. (2013). Human performance measures for interactive haptic-audio-visual interfaces. Haptics, IEEE Transactions on, 6(1), 46-57.

- (Jorda et al., 2005) Jorda, S., Kaltenbrunner, M., Geiger, G., & Bencina, R. (2005, August). The reactable*. In Proceedings of the international computer music conference (ICMC 2005), Barcelona, Spain (pp. 579-582).
- (Julien et al., 2010) Julien, C. A., Guastavino, C., Bouthillier, F., & Leide, J. E. (2010). Subject Explorer 3D: A Virtual Reality Collection Browsing and Searching Tool. In Proceedings of the 38th Annual Canadian Association for Information Science Conference, CAIS.
- (Kaaresoja et al., 2006) Kaaresoja, T., Brown, L. M., & Linjama, J. (2006, July). Snap-Crackle-Pop: Tactile feedback for mobile touch screens. In Proceedings of Eurohaptics (Vol. 2006, pp. 565-566).
- (Kanda et al., 2002) Kanda, T., Ishiguro, H., Ono, T., Imai, M., & Nakatsu, R. (2002). Development and evaluation of an interactive humanoid robot" Robovie". In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International

Conference on (Vol. 2, pp. 1848-1855). IEEE.

- (Kane et al, 2008) Kane, S. K., Bigham, J. P., & Wobbrock, J. O. (2008, October). Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (pp. 73-80). ACM.
- (Kanehisa et al., 2012) Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., & Tanabe, M. (2012). KEGG for integration and interpretation of large-scale molecular data sets. Nucleic acids research, 40(D1), D109-D114.
- (Kapec, 2010) Kapec, P. (2010, May). Visualizing software artifacts using hypergraphs. In Proceedings of the 26th Spring Conference on Computer Graphics (pp. 27-32). ACM.
- (Karpak et al., 2011) Karpak, B., Kasuganti, R. R., & Kumcu, E. (2011). Multi-objective decision-making in supplier selection: An application of visual interactive goal programming. Journal of Applied Business Research (JABR), 15(2), 57-72.
- (Kaschny et al., 2010) Kaschny, M., Buron, S., von Zadow, U., & Sostmann, K. (2010, November). Medical education on an interactive surface. In ACM International Conference on Interactive Table-tops and Surfaces (pp. 267-268). ACM.
- (Kato and Billinghurst, 1999) Kato, H., & Billinghurst, M. (1999). Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on (pp. 85-94). IEEE.
- (Kersting, 2012) Kersting, W. H. (2012). Distribution system modeling and analysis. CRC press.
- (Khoshelham et al., 2012) Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of Kinect depth data for indoor mapping applications. Sensors, 12(2), 1437-1454.
- (Khoshelham, 2011) Khoshelham, K. (2011, August). Accuracy analysis of Kinect depth data. In ISPRS workshop laser scanning (Vol. 38, No. 5, p. W12).
- (Kim and Maher, 2008) Kim, M. J., & Maher, M. L. (2008). The impact of tangible user interfaces on spatial cognition during collaborative design. Design Studies, 29(3), 222-253.
- (Kim et al., 2011) Kim, K., Proctor, R. W., & Salvendy, G. (2011). Comparison of 3D and 2D menus for cell phones. Computers in Human Behavior, 27(5), 2056-2066.

- (Kim et al., 2007) Kim, J., Park, J., Kim, H., & Lee, C. (2007, August). Hci (human computer interaction) using multi-touch table-top display. In Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on (pp. 391-394). IEEE.
- (Kin et al., 2009) Kin, K., Agrawala, M., & DeRose, T. (2009, May). Determining the benefits of direct-touch, bimanual, and multi finger input on a multi touch workstation. In Proceedings of Graphics interface 2009 (pp. 119-124). Canadian Information Processing Society.
- (Kobryn, 2000) Kobryn, C. (2000). Modeling components and frameworks with UML.
- (Koller et al., 1997) Koller, D., Klinker, G., Rose, E., Breen, D., Whitaker, R., & Tuceryan, M. (1997, September). Real-time vision-based camera tracking for augmented reality applications. In Proceedings of the ACM symposium on Virtual reality software and technology (pp. 87-94). ACM.
- (Kühn, 2011) Kühn, T. (2011). The kinect sensor platform. Advances In Media Technology, 1.
- (Kumar and Benbasat, 2004) Kumar, N., & Benbasat, I. (2004). The effect of relationship encoding, task type, and complexity on information representation: An empirical evaluation of 2D and 3D line graphs. Mis Quarterly, 255-281.
- (Lancaster et al., 2010) Lancaster, B. J., McNair, D. S., Parkins, K. D., & Yarbrough, M. E. (2010). U.S. Patent No. 7,707,045. Washington, DC: U.S. Patent and Trademark Office.
- (Lang et al., 2002) Lang, P., Kusej, A., Pinz, A., & Brasseur, G. (2002). Inertial tracking for mobile augmented reality. In Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE (Vol. 2, pp. 1583-1587). IEEE.
- (Leach et al., 2010) Leach, A. R., Gillet, V. J., Lewis, R. A., & Taylor, R. (2010). Three-dimensional pharmacophore methods in drug discovery. J. Med. Chem, 53(2), 539-558.
- (Lee, 2006) Lee, E. A. (2006). The problem with threads. Computer, 39(5), 33-42.
- (Lewis, 1995) Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. International Journal of Human-Computer Interaction, 7(1), 57-78.

- (Li, 2012) Li, Y. (2012, June). Hand gesture recognition using Kinect. In Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on (pp. 196-199). IEEE.

- (Likens, 2010) Likens, P. (2010). Art Review 2: Contemporary Digital Artifact. In Seminar ITGM705.
- (Linderman et al., 2008) Linderman, M. D., Collins, J. D., Wang, H., & Meng, T. H. (2008, March). Merge: a programming model for heterogeneous multi-core systems. In ACM SIGOPS operating systems review (Vol. 42, No. 2, pp. 287-296). ACM.

- (Lino et al., 2010) Lino, C., Christie, M., Lamarche, F., Schofield, G., & Olivier, P. (2010, July). A real-time cinematography system for interactive 3d environments. In Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (pp. 139-148). Eurographics Association.
- (Liu et al., 2010) Liu, Y., Qin, X., Xing, G., & Peng, Q. (2010). A new approach to outdoor illumination estimation based on statistical analysis for augmented reality. Computer Animation and Virtual Worlds, 21(3-4), 321-330.
- (Liu et al., 2011) Liu, T., Curtsinger, C., & Berger, E. D. (2011, October). Dthreads: efficient deterministic multithreading. In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 327-336). ACM.
- Lokesh et al., 2011) Lokesh, S., Balakrishnan, G., Malathy, S., & Murugan, K. (2010, July). Computer Interaction to human through photorealistic facial model for inter-process communication. In Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on (pp. 1-7). IEEE.
- (Ludwick and Doucette, 2009) Ludwick, D. A., & Doucette, J. (2009). Adopting electronic medical records in primary care: lessons learned from health information systems implementation experience in seven countries. International journal of medical informatics, 78(1), 22-31.
- (MacLaurin , 2011) MacLaurin, M. B. (2011, January). The design of Kodu: a tiny visual programming language for children on the Xbox 360. In ACM Sigplan Notices (Vol. 46, No. 1, pp. 241-246). ACM.

- (Mahatody et al., 2010) Mahatody, T., Sagar, M., & Kolski, C. (2010). State of the art on the cognitive walkthrough method, its variants and evolutions. Intl. Journal of Human–Computer Interaction, 26(8), 741-785.
- (Malik et al., 2005) Malik, S., Ranjan, A., & Balakrishnan, R. (2005, October). Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In Proceedings of the 18th annual ACM symposium on User

interface software and technology (pp. 43-52). ACM.

- (Malik, 2007) Malik, S. (2007). An exploration of multi-finger interaction on multi-touch surfaces (Doctoral dissertation, University of Toronto).
- (Marcus et al, 2003) Marcus, A., Feng, L., & Maletic, J. I. (2003, June). 3D representations for software visualization. In Proceedings of the 2003 ACM symposium on Software visualization (pp. 27-ff). ACM.
- (Marsh, 2011) Marsh, T. (2011) 'Serious games continuum: Between games for purpose and experiential environments for purpose', Entertainment Computing, 2, pp 61-68.
- (Martinet et al., 2010)Martinet, A., Casiez, G., & Grisoni, L. (2010, March). The design and evaluation of 3d positioning techniques for multi-touch displays. In 3D User Interfaces (3DUI), 2010 IEEE Symposium on (pp. 115-118). IEEE.
- (Matsushita and Rekimoto, 1997) Matsushita, N., & Rekimoto, J. (1997, October). HoloWall: designing a finger, hand, body, and object sensitive wall. In Proceedings of the 10th annual ACM symposium on User interface software and technology (pp. 209-210). ACM.
- (Mautz and Tilch, 2011) Mautz, R., & Tilch, S. (2011, September). Survey of optical indoor positioning systems. In Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on (pp. 1-7). IEEE.
- (Mautz, 2009) Mautz, R. (2009, March). The challenges of indoor environments and specification on some alternative positioning systems. In Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on (pp. 29-36). IEEE.
- (Maye et al., 2006) Maye, O., Schaeffner, J., & Maaser, M. (2006). An optical indoor positioning system for the mass market. In Proc. of the 3rd Workshop on Positioning, Navigation and Communication (pp. 111-116).
- (Mazalek, 2005) Mazalek, A. (2005). Media Tables: An extensible method for developing multi-user media interaction platforms for shared spaces (Doctoral dissertation, Massachusetts Institute of Technology).
- (Mehdi et al., 2013) Mehdi, M., Sahay, R., Derguech, W., & Curry, E. (2013, October). On-the-fly generation of multidimensional data cubes for web of things. In Proceedings of the 17th International Database Engineering & Applications Symposium (pp. 28-37). ACM.
- (Mehta, 1982) Mehta, N., Smith, K. C., & Holmes, F. E. (1982, May). Feature extraction as a tool for computer input. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82. (Vol. 7, pp. 818-820). IEEE.

- (Michalski and Grobelny, 2008) Michalski, R., & Grobelny, J. (2008). The role of colour preattentive processing in human–computer interaction task efficiency: a preliminary study. International Journal of Industrial Ergonomics, 38(3), 321-332.
- (Micire et al., 2011) Micire, M., Desai, M., Drury, J. L., McCann, E., Norton, A., Tsui, K. M., & Yanco, H. A. (2011, February). Design and validation of two-handed multi-touch tabletop controllers for robot teleoperation. In Proceedings of the 16th international conference on Intelligent user interfaces (pp. 145-154). ACM.
- (Mika and Greaves, 2012) Mika, P., & Greaves, M. (2012). Editorial: Semantic Web & Web 2.0. Web Semantics: Science, Services and Agents on the World Wide Web, 6(1).
- (Mistry and Maes, 2009) Mistry, P., & Maes, P. (2009, December). SixthSense: a wearable gestural interface. In ACM SIGGRAPH ASIA 2009 Sketches (p. 11). ACM.
- (Murugappan et al., 2013) Murugappan, S., Liu, H., & Ramani, K. (2013). Shape-It-Up: Hand gesture based creative expression of 3D shapes using intelligent generalized cylinders. Computer-Aided Design, 45(2), 277-287.
- (Myers et al., 2000) Myers, B., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface software tools. ACM Transactions on Computer-Human Interaction (TOCHI), 7(1), 3-28.
- (Nandi et al., 2013) Nandi, A., Jiang, L., Mandel, M., (2013). Gestural Query Specification. Proceedings of the VLDB Endowment, 7(4).
- (Nandi, 2013) Nandi, A. (2013). Querying Without Keyboards. In CIDR.
- (Neef and Ferranti, 2011) Naef, M., & Ferranti, E. (2011, March). Multi-touch 3D navigation for a building energy management system. In 3D User Interfaces (3DUI), 2011 IEEE Symposium on (pp. 113-114). IEEE.
- (Neumann et al., 2009) Neumann, C., Metoyer, R. A., & Burnett, M. (2009). End-user strategy programming. Journal of Visual Languages & Computing, 20(1), 16-29.
- (Nickolls and Dally, 2010) Nickolls, J., & Dally, W. J. (2010). The GPU computing era. IEEE micro, 30(2), 56-69.
- (Nickolls et al., 2008) Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable parallel programming with CUDA. Queue, 6(2), 40-53.
- (Nielsen, 1994) Nielsen, J. (1994, April). Usability inspection methods. In Conference companion on Human factors in computing systems (pp. 413-414). ACM.

- (Nishino et al., 1997) Nishino, H., Utsumiya, K., Kuraoka, D., Yoshioka, K., & Korida, K. (1997, September). Interactive two-handed gesture interface in 3D virtual environments. In Proceedings of the ACM symposium on Virtual reality software and technology (pp. 1-8). ACM.
- (Noguera and Torres, 2013) Noguera, J. M., & Torres, J. C. (2013). Interaction and visualization of 3D virtual environments on mobile devices. Personal and ubiquitous computing, 17(7), 1485-1486.
- (North et al., 2009) North, C., Dwyer, T., Lee, B., Fisher, D., Isenberg, P., Robertson, G., & Inkpen, K. (2009). Understanding multi-touch manipulation for surface computing. In Human-Computer Interaction–INTERACT 2009 (pp. 236-249). Springer Berlin Heidelberg.
- (Oikonomidis et al., 2011) Oikonomidis, I., Kyriazis, N., & Argyros, A. A. (2011, August). Efficient model-based 3D tracking of hand articulations using Kinect. In BMVC (pp. 1-11).
- (Okamura et al., 2004) Okamura, T., Shizuki, B., & Tanaka, J. (2004, July). Execution visualization and debugging in three-dimensional visual programming. In Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on (pp. 167-172). IEEE.
- (Oldewurtel et al., 2010) Oldewurtel, F., Gyalistras, D., Gwerder, M., Jones, C. N., Parisio, A., Stauch, V., & Morari, M. (2010, May). Increasing energy efficiency in building climate control using weather forecasts and model predictive control. In 10th REHVA World Congress Clima (pp. 9-12).
- (Olson, 1998) Olson, I. C., Atrash Leong, Z., Wilensky, U., & Horn, M. S. (2011, January). It's just a toolbar!: using tangibles to help children manage conflict around a multi-touch tabletop. In Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction (pp. 29-36). ACM.
- (Olsson et al., 2012) Olsson, T., Kärkkäinen, T., Lagerstam, E., & Ventä-Olkkonen, L. (2012). User evaluation of mobile augmented reality scenarios. Journal of Ambient Intelligence and Smart Environments, 4(1), 29-47.
- (Osawa, 2006) Osawa, N. (2006, June). Automatic adjustments for efficient and precise positioning and release of virtual objects. In Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications (pp. 121-128). ACM.
- (Pavlovic et al., 1997) Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. Pattern

Analysis and Machine Intelligence, IEEE Transactions on, 19(7), 677-695.

- (Peltonen et al., 2008) Peltonen, P., Kurvinen, E., Salovaara, A., Jacucci, G., Ilmonen, T., Evans, J., ... & Saarikko, P. (2008, April). It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1285-1294). ACM.
- (Peters, 2004) Peters, J. F. (2004). Approximation space for intelligent system design patterns. Engineering Applications of Artificial Intelligence, 17(4), 393-400.
- (Petrie and Power, 2012) Petrie, H., & Power, C. (2012, May). What do users really care about?: a comparison of usability problems found by users and experts on highly interactive websites. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 2107-2116). ACM.
- (Pfeiffer and Franke, 2011) Pfeiffer, D., & Franke, U. (2011, August). Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In BMVC (pp. 1-12).
- (Pino et al., 2013) Pino, A., Tzemis, E., Ioannou, N., & Kouroupetroglou, G. (2013). Using Kinect for 2D and 3D pointing tasks: performance evaluation. In Human-Computer Interaction. Interaction Modalities and Techniques (pp. 358-367). Springer Berlin Heidelberg.
- (Ponto et al., 2011) Ponto, K., Doerr, K., Wypych, T., Kooker, J., & Kuester, F. (2011). CGLXTouch: A multi-user multi-touch approach for ultra-high-resolution collaborative workspaces. Future Generation Computer Systems, 27(6), 649-656.
- (Poole, 2004) Poole, S. (2004). Trigger happy: Videogames and the entertainment revolution. Skyhorse Publishing Inc.
- (Poupyrev et al., 2007) Poupyrev, I., Nashida, T., & Okabe, M. (2007, February). Actuation and tangible user interfaces: the Vaucanson duck, robots, and shape displays. In Proceedings of the 1st international conference on Tangible and embedded interaction (pp. 205-212). ACM.
- (Qian et al., 2011) Qian, H., Kuber, R., Sears, A., & Murphy, E. (2011). Maintaining and modifying pace through tactile and multimodal feedback. Interacting with Computers, 23(3), 214-225.
- (Raheja et al., 2011) Raheja, J. L., Chaudhary, A., & Singal, K. (2011, September). Tracking of fingertips and centers of palm using kinect. In Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on (pp. 248-252). IEEE.
- (Ratti et al., 2004) Ratti, C., Wang, Y., Ishii, H., Piper, B., & Frenchman, D. (2004).

Tangible User Interfaces (TUIs): a novel paradigm for GIS. Transactions in GIS, 8(4), 407-421.

- (Ray et al., 2007) Ray, A., & Bowman, D. A. (2007, November). Towards a system for reusable 3D interaction techniques. In Proceedings of the 2007 ACM symposium on Virtual reality software and technology (pp. 187-190). ACM.
- (Regenbrecht, Baratoff and Wagner, 2001) Regenbrecht, H. T., Wagner, M., & Baratoff, G. (2002). Magicmeeting: A collaborative tangible augmented reality system. Virtual Reality, 6(3), 151-166.
- (Reiss, 2007) Reiss, S. P. (2007). Visual representations of executing programs. Journal of Visual Languages & Computing, 18(2), 126-148.
- (Ren and O'Neill, 2012) Ren, G., & O'Neill, E. (2012, March). 3D marking menu selection with freehand gestures. In 3D User Interfaces (3DUI), 2012 IEEE Symposium on (pp. 61-68). IEEE.
- (Ren et al., 2011) Ren, Z., Meng, J., Yuan, J., & Zhang, Z. (2011, November). Robust hand gesture recognition with kinect sensor. In Proceedings of the 19th ACM international conference on Multimedia (pp. 759-760). ACM.
- (Ren et al., 2013) Ren, Z., Yuan, J., Meng, J., & Zhang, Z. (2013). Robust part-based hand gesture recognition using kinect sensor. IEEE Trans. Multimedia, 15(5), 1110-1120.
- (Rivera-Borroto et al., 2011) Rivera-Borroto, O. M., Marrero-Ponce, Y., García-de la Vega, J. M., & Grau-A´balo, R. D. C. (2011). Comparison of combinatorial clustering methods on pharmacological data sets represented by machine learning-selected real molecular descriptors. Journal of chemical information and modeling, 51(12), 3036-3049.
- (Ronfard and Taubin, 2007) Ronfard, R., & Taubin, G. (2007). Introducing 3D Cinematography [Guest editors' introduction]. Computer Graphics and Applications, IEEE, 27(3), 18-20.
- (Rosner et al., 2006) Rosner, B., Glynn, R. J., & Lee, M. L. T. (2006). The Wilcoxon signed rank test for paired comparisons of clustered data. Biometrics, 62(1), 185-192.
- (Rotard et al., 2005) Rotard, M., Weiskopf, D., & Ertl, T. (2005). A combined introductory course on human–computer interaction and computer graphics. Computers & Graphics, 29(2), 267-272.
- (Roy and Pattnaik 2014) Roy, S., & Pattnaik, P. K. (2014, January). Some Popular Usability Evaluation Techniques for Websites. In Proceedings of the International

Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013 (pp. 535-543). Springer International Publishing.

- (Rozier, 2000) Rozier, J. M. (2000). Hear&There: An augmented reality system of linked audio (Doctoral dissertation, Massachusetts Institute of Technology).
- (Rukzio et al., 2009) Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., & Chin, J. (2006). An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In UbiComp 2006: Ubiquitous Computing (pp. 87-104). Springer Berlin Heidelberg.
- (Ryall et al., 2006) Ryall, K., Morris, M. R., Everitt, K., Forlines, C., & Shen, C. (2006, January). Experiences with and observations of direct-touch tabletops. In Horizontal Interactive Human-Computer Systems, 2006. TableTop 2006. First IEEE International Workshop on (pp. 8-pp). IEEE.
- (Ryoo et al., 2008) Ryoo, S., Rodrigues, C. I., Baghsorkhi, S. S., Stone, S. S., Kirk, D. B., & Hwu, W. M. W. (2008, February). Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming (pp. 73-82). ACM.
- (Saffer, 2009) Saffer, D. (2008). Designing gestural interfaces: Touchscreens and interactive devices. " O'Reilly Media, Inc.".
- Schaefer et al., 2009) Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., & Buetow, K. H. (2009). PID: the pathway interaction database. Nucleic acids research, 37(suppl 1), D674-D679.
- (Schlömer et al., 2008) Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008, February). Gesture recognition with a Wii controller. In Proceedings of the 2nd international conference on Tangible and embedded interaction (pp. 11-14). ACM.
- (Schmid et al., 2010) Schmid, B., Schindelin, J., Cardona, A., Longair, M., & Heisenberg, M. (2010). A high-level 3D visualization API for Java and ImageJ. BMC bioinformatics, 11(1), 274.
- (Sears and Jacko, 2007) Sears, A., & Jacko, J. A. (Eds.). (2007). The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications. CRC press.(Chapter 1)
- (Segen and Kumar, 1998) Segen, J., & Kumar, S. (1998, September). Gesture vr: vision-based 3d hand interace for spatial interaction. In Proceedings of the sixth ACM international conference on Multimedia (pp. 455-464). ACM.
- (Shen et al., 2005) Shen, F., Kang, L., Fang, B., & Wu, Z. The pen-computer interface based on force tablet: active information acquisition for human-computer-

interaction. In Information Acquisition, 2005 IEEE International Conference on (pp. 6-pp). IEEE.

- (Shim et al., 2005) Shim, H., & Fishwick, P. (2005, December). Programming using dynamic system modeling VIA a 3D-based multimodeling framework. In Proceedings of the 37th conference on Winter simulation (pp. 1931-1938). Winter Simulation Conference.
- Sirazzi et al., 2009) Shirazi, A. S., Döring, T., Parvahan, P., Ahrens, B., & Schmidt, A. (2009, September). Poker surface: combining a multi-touch table and mobile phones in interactive card games. In Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (p. 73). ACM.
- (Simoes et al., 2013) Simoes, F. P. M., Roberto, R. A., Figueiredo, L. S., Lima, J. P. S. D. M., Almeida, M. W., & Teichrieb, V. (2013, May). 3D Tracking in Industrial Scenarios: A Case Study at the ISMAR Tracking Competition. In Virtual and Augmented Reality (SVR), 2013 XV Symposium on (pp. 97-106). IEEE.
- (Smisek et al., 2013) Smisek, J., Jancosek, M., & Pajdla, T. (2013). 3D with Kinect. In Consumer Depth Cameras for Computer Vision (pp. 3-25). Springer London.
- (Sodhi et al., 2013) Sodhi, R. S., Jones, B. R., Forsyth, D., Bailey, B. P., & Maciocci, G. (2013, April). BeThere: 3D mobile collaboration with spatial input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 179-188). ACM.
- (Stefanidis et al., 2011) Stefanidis, K., Koutrika, G., & Pitoura, E. (2011). A survey on representation, composition and application of preferences in database systems. ACM Transactions on Database Systems (TODS), 36(3), 19.
- (Steffin, 1999) Steffin, M. (1999). Visual-haptic interfaces: Modification of motor and cognitive performance. Neurology.
- (Stodle et al., 2009) Stodle, D., Troyanskaya, O., Li, K., & Anshus, O. J. (2009, March). Tech-note: Device-free interaction spaces. In 3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on (pp. 39-42). IEEE.
- (Sturman and Zeltzer, 1994) Sturman, D. J., & Zeltzer, D. (1994). A survey of glove-based input. Computer Graphics and Applications, IEEE, 14(1), 30-39.
- (Sutherland, 1968) Sutherland, I. E. (1968, December). A head-mounted three dimensional display. In Proceedings of the December 9-11, 1968, fall joint computer conference, part I (pp. 757-764). ACM.

- (Takala et al., 2012) Takala, T. M., Rauhamaa, P., & Takala, T. (2012, March). Survey of 3DUI applications and development challenges. In 3D User Interfaces (3DUI), 2012 IEEE Symposium on (pp. 89-96). IEEE.
- (Takeoka, Miyaki and Rekimoto, 2010) Takeoka, Y., Miyaki, T., & Rekimoto, J. (2010, November). Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In ACM International Conference on Interactive Tabletops and Surfaces (pp. 91-94). ACM.
- (Tang, 2011) Tang, M. (2011). Recognizing hand gestures with microsoft's kinect. Palo Alto: Department of Electrical Engineering of Stanford University:[sn].
- (Tanimoto et al., 2011) Tanimoto, M., Tehrani, M. P., Fujii, T., & Yendo, T. (2011). Free-viewpoint TV. Signal Processing Magazine, IEEE, 28(1), 67-76.
- (Tanskanen et al., 2013) Tanskanen, P., Kolev, K., Meier, L., Camposeco, F., Saurer, O., & Pollefeys, M. (2013, December). Live metric 3d reconstruction on mobile phones. In Computer Vision (ICCV), 2013 IEEE International Conference on (pp. 65-72). IEEE.
- (Teather and Stuerzlinger, 2011) Teather, R. J., & Stuerzlinger, W. (2011, March). Pointing at 3D targets in a stereo head-tracked virtual environment. In 3D User Interfaces (3DUI), 2011 IEEE Symposium on (pp. 87-94). IEEE.
- (Tekli et al., 2013) Tekli, G., Chbeir, R., & Fayolle, J. (2013). A visual programming language for XML manipulation. Journal of Visual Languages & Computing, 24(2), 110-135.
- (Telea and Voinea, 2011) Telea, A., & Voinea, L. (2011). Visual software analytics for the build optimization of large-scale software systems. Computational Statistics, 26(4), 635-654.
- (Telea et al., 2010) Telea, A. C., Voinea, L., & Sassenburg, H.(2010) Visual Tools for Software Architecture Understanding. IEEE Software, 27(6), pp. 46-53.
- (Teyseyre et al., 2009)Teyseyre, A. R., & Campo, M. R. (2009). An overview of 3D software visualization. Visualization and Computer Graphics, IEEE Transactions on, 15(1), 87-105.
- (Tse et al., 2006) Tse, E., Shen, C., Greenberg, S., & Forlines, C. (2006, May). Enabling interaction with single user applications through speech and gestures on a multi-user tabletop. In Proceedings of the working conference on Advanced visual interfaces (pp. 336-343). ACM.
- (Ullmer and Ishii, 1997) Ullmer, B., & Ishii, H. (1997, October). The metaDESK: models and prototypes for tangible user interfaces. In Proceedings of the 10th annual ACM symposium on User interface software and technology (pp. 223-232).

ACM.

- (Underkoffler and Ishii, 1998) Underkoffler, J., & Ishii, H. (1998, January). Illuminating light: an optical design tool with a luminous-tangible interface. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 542-549). ACM Press/Addison-Wesley Publishing Co.
- (Van den Bergh et al., 2011) Van den Bergh, M., Carton, D., De Nijs, R., Mitsou, N., Landsiedel, C., Kuehnlenz, K.,& Buss, M. (2011, July). Real-time 3D hand gesture interaction with a robot for understanding directions from humans. In RO-MAN, 2011 IEEE (pp. 357-362). IEEE.
- (Van den Bergh and Van Gool, 2011) Van den Bergh, M., & Van Gool, L. (2011, January). Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In Applications of Computer Vision (WACV), 2011 IEEE Workshop on (pp. 66-72). IEEE.
- (Van Krevelen and Poelman, 2010) Van Krevelen, D. W. F., & Poelman, R. (2010). A survey of augmented reality technologies, applications and limitations. International Journal of Virtual Reality, 9(2), 1.
- (Vassilidis, 1998) Vassiliadis, P. (1998, July). Modeling multidimensional databases, cubes and cube operations. In Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on (pp. 53-62). IEEE.
- (Vazquez et al., 2010) Vazquez, G., Andres Diaz Pace, J., & Campo, M. (2010). Reusing design experiences to materialize software architectures into object-oriented designs. Information Sciences.
- (Velenis et al., 2009) Velenis, D., Stucchi, M., Marinissen, E. J., Swinnen, B., & Beyne, E. (2009, September). Impact of 3D design choices on manufacturing cost. In 3D System Integration, 2009. 3DIC 2009. IEEE International Conference on (pp. 1-5). IEEE.
- (Vicknair et al., 2010) Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010, April). A comparison of a graph database and a relational database: a data provenance perspective. In Proceedings of the 48th annual Southeast regional conference (p. 42). ACM.
- (Villar et al., 2009) Villar, N., Izadi, S., Rosenfeld, D., Benko, H., Helmes, J., Westhues, J. & Chen, B. (2009, October). Mouse 2.0: multi-touch meets the mouse. In Proceedings of the 22nd annual ACM symposium on User interface software and technology (pp. 33-42). ACM.

- (Vogel and Balakrishnan, 2005) Vogel, D., & Balakrishnan, R. (2005, October). Distant freehand pointing and clicking on very large, high resolution displays. In Proceedings of the 18th annual ACM symposium on User interface software and technology (pp. 33-42). ACM.
- (Volk et al., 2011) Volk, T., Althoff, F., Lang, M., & Grahn, H. (2011, August). A Generic User Interface Framework. In Proc. Of the 9th Int. Conf. on Human-Computer Interaction (pp. 1-9).
- (von Pilgrim and Duske, 2008) von Pilgrim, J., & Duske, K. (2008, September). Gef3D: a framework for two-, two-and-a-half-, and three-dimensional graphical editors. In Proceedings of the 4th ACM symposium on Software visualization (pp. 95-104). ACM.
- (Wachs et al., 2011) Wachs, J. P., Kölsch, M., Stern, H., & Edan, Y. (2011). Vision-based hand-gesture applications. Communications of the ACM, 54(2), 60-71.
- (Wagner et al., 2009) Wagner, I., Basile, M., Ehrenstrasser, L., Maquil, V., Terrin, J. J., & Wagner, M. (2009, June). Supporting community engagement in the city: urban planning in the MR-tent. In Proceedings of the fourth international conference on Communities and technologies (pp. 185-194). ACM.
- (Wagner et al., 2010) Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T., & Schmalstieg, D. (2010). Real-time detection and tracking for augmented reality on mobile phones. Visualization and Computer Graphics, IEEE Transactions on, 16(3), 355-368.
- (Wagner et al., 2012) Wagner, J., Huot, S., & Mackay, W. (2012, May). Bitouch and bipad: Designing bimanual interaction for hand-held tablets. In Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (pp. 2317-2326). ACM.
- (Wagner et al., 2013) Wagner, R., Piovesan, S. D., Passerino, P., Maria, L., & de Lima, J. V. (2013, October). Using 3D virtual learning environments in new perspective of education. In Information Technology Based Higher Education and Training (ITHET), 2013 International Conference on (pp. 1-6). IEEE.
- (Wang et al., 2007) Wang, P. H., Collins, J. D., Chinya, G. N., Jiang, H., Tian, X., Girkar, M., ... & Wang, H. (2007, June). EXOCHI: architecture and programming environment for a heterogeneous multi-core multithreaded system. In ACM SIGPLAN Notices (Vol. 42, No. 6, pp. 156-166). ACM.
- (Weng et al., 2013) Weng, E. N. G., Khan, R. U., Adruce, S. A. Z., & Bee, O. Y. (2013). Objects Tracking from Natural Features in Mobile Augmented Reality. Procedia-Social and Behavioral Sciences, 97, 753-760.

- (Westerman and Elias, 2001) Westerman, W., & Elias, J. G. (2001). U.S. Patent No. 6,323,846. Washington, DC: U.S. Patent and Trademark Office.
- (Wigdor and Wixon, 2011) Wigdor, D., & Wixon, D. (2011). Brave NUI world: designing natural user interfaces for touch and gesture. Elsevier.(chapter 2)
- (Wigdor et al., 2007) Wigdor, D., Perm, G., Ryall, K., Esenther, A., & Shen, C. (2007, October). Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In Horizontal Interactive Human-Computer Systems, 2007. TABLETOP'07. Second Annual IEEE International Workshop on (pp. 60-67). IEEE.
- (Wilson, 2004) Wilson, A. D. (2004, October). TouchLight: an imaging touch screen and display for gesture-based interaction. In Proceedings of the 6th international conference on Multimodal interfaces (pp. 69-76). ACM.
- (Wilson, 2005) Wilson, A. D. (2005, October). PlayAnywhere: a compact interactive tabletop projection-vision system. In Proceedings of the 18th annual ACM symposium on User interface software and technology (pp. 83-92). ACM.
- (Wingrave et al., 2010) Wingrave, C. A., Williamson, B., Varcholik, P. D., Rose, J., Miller, A., Charbonneau, E. & LaViola, J. J. (2010). The wiimote and beyond: Spatially convenient devices for 3d user interfaces. Computer Graphics and Applications, IEEE, 30(2), 71-85.
- (Wolter, 2012) Wolter, J. (2012, August). DEViL3D-A Generator Framework for Three-Dimensional Visual Languages. In DMS (pp. 171-176).
- (Wong et al., 2012) Wong, Y. W., Dubrownik, T., Tang, W. T., Tan, W. J., Duan, R., Goh, R. S. M., ... & Wong, W. F. (2012). Tulipse: a visualization framework for user-guided parallelization. In Euro-Par 2012 Parallel Processing (pp. 4-15). Springer Berlin Heidelberg.
- (Wu and Balakrishnan, 2003) Wu, M., & Balakrishnan, R. (2003, November). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In Proceedings of the 16th annual ACM symposium on User interface software and technology (pp. 193-202). ACM.
- (Wu et al., 2005) Wu, M., & Balakrishnan, R. (2003, November). Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In Proceedings of the 16th annual ACM symposium on User interface software and technology (pp. 193-202). ACM.
- (Wurm et al., 2010) Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010, May). OctoMap: A probabilistic, flexible, and compact 3D map

representation for robotic systems. In Proc. of the ICRA 2010 workshop on best practice in 3D perception and modeling for mobile manipulation (Vol. 2).

- (Xavier and Araujo, 2008) Xavier, J., & Araújo, H. (2008). The Experimental Robotics Framework. In Simulation, Modeling, and Programming for Autonomous Robots (pp. 207-221). Springer Berlin Heidelberg.
- (Xia et al., 2011) Xia, L., Chen, C. C., & Aggarwal, J. K. (2011, June). Human detection using depth information by kinect. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on (pp. 15-22). IEEE.
- (Yasuda, 1999) Yasuda, G. I. (1999). An object-oriented multitasking control environment for multirobot system programming and execution with 3D graphic simulation. International journal of production economics, 60, 241-250.
- (You et al., 2001) You, S., & Neumann, U. (2001, March). Fusion of vision and gyro tracking for robust augmented reality registration. In Virtual Reality, 2001. Proceedings. IEEE (pp. 71-78). IEEE.
- (Zhang et al., 2013) Zhang, J. K., Ma, C. X., Liu, Y. J., Fu, Q. F., & Fu, X. L. (2013). Collaborative Interaction for Videos on Mobile Devices Based on Sketch Gestures. Journal of Computer Science and Technology, 28(5), 810-817.
- (Zhao et al., 2011) Zhao, P., Li, X., Xin, D., & Han, J. (2011, June). Graph cube: on warehousing and OLAP multidimensional networks. In Proceedings of the 2011 ACM SIGMOD International Conference on Management of data (pp. 853-864). ACM.
- (Zhou et al., 2008) Zhou, F., Duh, H. B. L., & Billinghurst, M. (2008, September). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (pp. 193-202). IEEE Computer Society.

Appendix

A. Usability questionnaires of experiments chapter 3

Usability Survey Touch Game

Number		**Subject ID**	

Age:	**Gender:**

Experiment 1: Game with AR Enable

Time to perform the Task:	
HITS	
MISSES	

Scale: ***Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.***

Questions	**1**	**2**	**3**	**4**	**5**
Was the interaction easy to understand?					
Was it easy to manipulate?					
Is the navigation system intuitive?					
How would you rate:					
The Interface?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					
The commands selected (i.e. tap, move around)?					

Experiment 2: Game with AR disabled

Time to perform the Task:	
HITS	
MISSES	

Scale: ***Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.***

Questions	1	2	3	4	5
Was the interaction easy to understand?					
Was it easy to manipulate?					
Is the navigation system intuitive?					
How would you rate:					
The Interface?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					
The commands selected (i.e. tap, touch rotation)?					

Usability Survey 3D Touch game cooperative version

Couple Number	

Subject ID	

Age:	**Gender:**

Experiment 1: Game with AR Enable

Time to perform the Task:	
HITS	
MISSES	

Scale: *Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.*

Questions	**1**	**2**	**3**	**4**	**5**
The feedback from the other user's actions?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					

Experiment 2: Game with AR disabled

Time to perform the Task:	
HITS	
MISSES	

Scale: *Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.*

Questions	**1**	**2**	**3**	**4**	**5**
The feedback from the other user's actions?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					

B. Usability questionnaires of experiments chapter 4 and 5

Usability Survey Cube Interaction

Number	

The Interviewed knows SQL:	
Age:	**Gender:**

Experiment 1: Cube Click selection

Scale: ***Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.***

Questions	**1**	**2**	**3**	**4**	**5**
Was the interaction easy to understand?					
Was it easy to manipulate?					
Is the navigation system intuitive?					
How would you rate:					
The Interface?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					
The hand gestures selected?					
SQL interface compared with the proposed visual approach (see page 3)					
The selection is easier than SQL?					
The task is more intuitive than SQL sentences?					
Is it easier to learn the proposed visual approach than SQL?					
Is the task faster to perform than with SQL?					

Experiment 2: Cube slide selection

Scale: ***Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.***

Questions	**1**	**2**	**3**	**4**	**5**
Was the interaction easy to understand?					
Was it easy to manipulate?					
Is the navigation system intuitive?					
How would you rate:					
The Interface?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					
The hand gestures selected?					
SQL interface compared with the proposed visual approach (see page 4)					
The selection is easier than SQL?					
The task is more intuitive than SQL sentences?					
Is it easier to learn the proposed visual approach than SQL?					
Is the task faster to perform than with SQL?					

Final Question:
In order of performance and simplicity give places, 1(most preferable) to 3 (less preferable to each approach:

	1	**2**	**3**
SQL Normal Approach:			
Kinect Approach 1:			
Kinect Approach2:			

Experiment 3: 3D Multi-thread interface

Scale: ***Extremely bad = 1; Bad = 2; Normal=3; Good=4; Extremely good=5.***

Questions	1	2	3	4	5
Was the interaction easy to understand?					
Was it easy to manipulate?					
Is the navigation system intuitive?					
How would you rate:					
The Interface?					
The Performance?					
The functionality?					
The objective achieved?					
The user experience?					
The hand gestures selected?					

Final Question:
In order of performance and simplicity give places, tick your most preferable approach (see page 5).

APPROACH	
C++ Normal Approach:	
Kinect 3D Multi-thread interface	

TASK PERFORMED WITH SQL

TABLES:

PATIENT ID		
ID Code	**Name**	**Gender**
AB101	Bloggs	Male
RS102	Smith	Female
GW103	Williams	Female
MD104	Doe	Male

WEIGHT			
ID Code	**July**	**August**	**September**
AB101	71	69	68
RS102	46	44	42
GW103	12	12	14
MD104	87	86	86

SQL QUERY:

```
SELECT patient_ID.IDCode,
weight.july, weight.august

FROM patient JOIN weight

ON (patient_ID.idcode =
weight.idcode)
```

RESULT:

ID Code	July	August
AB101	71	69
RS102	46	44
GW103	12	12
MD104	87	86

TASK PERFORMED WITH C++ CODE

```
#include <windows.h>
#include <iostream>
#include <thread>
#include <mutex>
#include <ctime>

#define LL 100000000

using namespace std;
const int REPS = 1;
double global_sum=0.0;
double aa[LL];
mutex mutex_sum;

void sum(int TID, int N, int NT)
{
      long i0 = TID*(N/NT), i1 = i0 + (N/NT);
      double localSum=0.0;
      for (int r =0; r<REPS; r++)
            for (int i=i0; i<i1; i++)
                  localSum += aa[i];
      mutex_sum.lock();
      global_sum += localSum;
      mutex_sum.unlock();
}

int main()
{
      for(int t=0; t<LL; t++)
      {
            aa[t]=t+1;
      }

      thread *thrds = new thread[4];
      int NT=4;
      for (int t=0; t<NT; t++)
      {
            thrds[t] = thread(sum, t, LL, NT);
      }
      for (int t=0; t<NT; t++)
      {
            thrds[t].join();
      }
      cout << "Thread: Sum = "<< global_sum << endl;
      return 0;
}
```

C. Interaction thresholds for interface presented in Chapter 4

Threshold definition for each action.

Gesture	**Threshold Height (vertical axis)**	**Threshold Width (horizontal axis)**	**Threshold Depth (parallel to screen axis)**
Rotation	-	*R/X*	-
Clicking	*U/Y*	*U/Y*	*K/I*
Swiping	*V/Y*	*U/X*	-

Numerical threshold's values for each gesture defined.

Gesture	**Threshold vertical axis (%)**	**Threshold horizontal axis (%)**	**Threshold depth axis (%)**
Rotation	-	7.2%	-
Clicking	9.1%	9.1%	10.0%
Swiping	27.3%	9.1%	-

D. Interaction thresholds for interface presented in Chapter 5

Threshold definition for each action.

Gesture	Threshold Height (vertical axis)	Threshold Width (horizontal axis)
Rotation	-	*R/X*
Grabbing	*E/Y*	*E/X*
Placing	*E/Y*	*E/X*

Numerical threshold's values for each gesture defined.

Gesture	Threshold Height (vertical axis)	Threshold Width (horizontal axis)
Rotation	-	*40%*
Grabbing	*20%*	*15%*
Placing	*20%*	*15%*