



UNIVERSITY OF CAGLIARI

PHD SCHOOL OF MATHEMATICS AND  
COMPUTER SCIENCE

Ciclo XXVI - PhD course of Computer Science - SSD: INF/01

---

# Interactive Spaces

Natural interfaces supporting  
gestures and manipulations in interactive spaces

---

*Author:*

Samuel Aldo IACOLINA

*PhD coordinator:*

Prof. Michele PINNA

*Supervisor:*

Prof. Riccardo SCATENI

ACADEMIC YEAR: 2012-2013



---

## Abstract

This doctoral dissertation focuses on the development of interactive spaces through the use of natural interfaces based on gestures and manipulative actions. In the real world people use their senses to perceive the external environment and they use manipulations and gestures to explore the world around them, communicate and interact with other individuals. From this perspective the use of natural interfaces that exploit the human sensorial and explorative abilities helps filling the gap between physical and digital world.

In the first part of this thesis we describe the work made for improving interfaces and devices for tangible, multi-touch and free-hand interactions. The idea is to design devices able to work also in uncontrolled environments, and in situations where control is mostly of the physical type where even the less experienced users can express their manipulative exploration and gesture communication abilities.

We also analyze how it can be possible to mix these techniques to create an interactive space, specifically designed for teamwork where the natural interfaces are distributed in order to encourage collaboration.

We then give some examples of how these interactive scenarios can host various types of applications facilitating, for instance, the exploration of 3D models, the enjoyment of multimedia contents and social interaction.

Finally we discuss our results and put them in a wider context, focusing our attention particularly on how the proposed interfaces actually improve people's lives and activities and the interactive spaces become a place of aggregation where we can pursue objectives that are both personal and shared with others.

An accompanying playlist of videos is available at <http://www.youtube.com/playlist?list=PLHyH1JK4RRtkh9d09EtoP34vAFtrq1o40>. Where present, the background music is a personal composition of the author.



# Contents

Abstract . . . . .	iii
<b>I Background and Motivations</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Scope and Goals . . . . .	4
1.2 Dissertation Structure . . . . .	5
<b>2 Motivations</b>	<b>7</b>
<b>II Related Research</b>	<b>9</b>
<b>3 Towards Natural Interfaces</b>	<b>11</b>
3.1 Interfaces evolution . . . . .	11
3.2 HCI revolution . . . . .	14
<b>4 Manipulative and Gestural Interfaces</b>	<b>17</b>
4.1 What is Natural . . . . .	17
4.2 Chosen Interfaces . . . . .	19
4.3 Gestural Interaction . . . . .	20
4.4 Tangible User Interfaces . . . . .	21
4.5 Multitouch Interfaces . . . . .	23
4.6 Free-Hand Interfaces . . . . .	25
4.7 Objectives Updated . . . . .	28
<b>III How to set-up an Interactive Space</b>	<b>31</b>
<b>5 Tangible Interaction</b>	<b>33</b>

5.1	Choosing the appropriate application . . . . .	33
5.2	Interactive Table . . . . .	35
5.3	Evaluation and Discussion . . . . .	37
5.4	Summarising . . . . .	38
<b>6</b>	<b>Multi-touch Sensors Improvement</b>	<b>39</b>
6.1	Multitouch Table . . . . .	40
6.2	Converting showcases and media facades into interactive walls .	46
6.3	Summarising . . . . .	53
<b>7</b>	<b>Free-Hand Interaction</b>	<b>55</b>
7.1	Natural exploration of 3D models . . . . .	55
7.2	3D Interaction . . . . .	56
7.3	Our Proposal . . . . .	58
7.4	Summarising . . . . .	61
<b>8</b>	<b>A Unifying Framework</b>	<b>63</b>
8.1	Goals . . . . .	63
8.2	Description . . . . .	64
8.3	Implementation details . . . . .	66
8.4	Discussion and summarising . . . . .	66
<b>IV</b>	<b>Gestural Interaction</b>	<b>69</b>
<b>9</b>	<b>Analyzing the Gestural Action</b>	<b>71</b>
9.1	Browsing visual documents by free-hand gestures . . . . .	71
9.2	Web based Video Annotation . . . . .	74
<b>10</b>	<b>Cooperation</b>	<b>81</b>
10.1	Manipulative and gestural experience . . . . .	81
10.2	An improved OCGM GUI . . . . .	83
10.3	Multi-user management . . . . .	86
10.4	Preliminary tests . . . . .	88
10.5	Discussion and summarising . . . . .	88
<b>11</b>	<b>Evaluation</b>	<b>91</b>
11.1	Evaluation of gestures in multi-touch interaction . . . . .	92
11.2	Comparison between multi-touch and free-hand interaction . .	104
<b>12</b>	<b>Case study: Architecture and Construction</b>	<b>113</b>

12.1	Background . . . . .	113
12.2	Interacting with massive point clouds . . . . .	114
12.3	Multi-resolution Solid Images . . . . .	115
12.4	Exploring massive point clouds using multi-touch gestures . . .	128
12.5	Free-hand interaction supporting volume calculation of digital elevation models . . . . .	129
12.6	Summarising . . . . .	135
 <b>V Final Remarks</b>		<b>137</b>
 <b>13 Conclusions</b>		<b>139</b>
13.1	Contributions . . . . .	142
13.2	Fairs and Exhibits . . . . .	144
13.3	Published as . . . . .	146
 <b>14 Personal Notes</b>		<b>149</b>
 <b>Appendices</b>		<b>151</b>
 <b>A “Seeing” the Music</b>		<b>153</b>
A.1	A “musical” lamp . . . . .	153
A.2	Multi-touch music player . . . . .	155
 <b>B Human Computer Interaction</b>		<b>157</b>
B.1	Main objectives . . . . .	157
B.2	Interfaces . . . . .	161
B.3	Designing Interfaces . . . . .	162
B.4	Evaluation . . . . .	162
 <b>References</b>		<b>165</b>





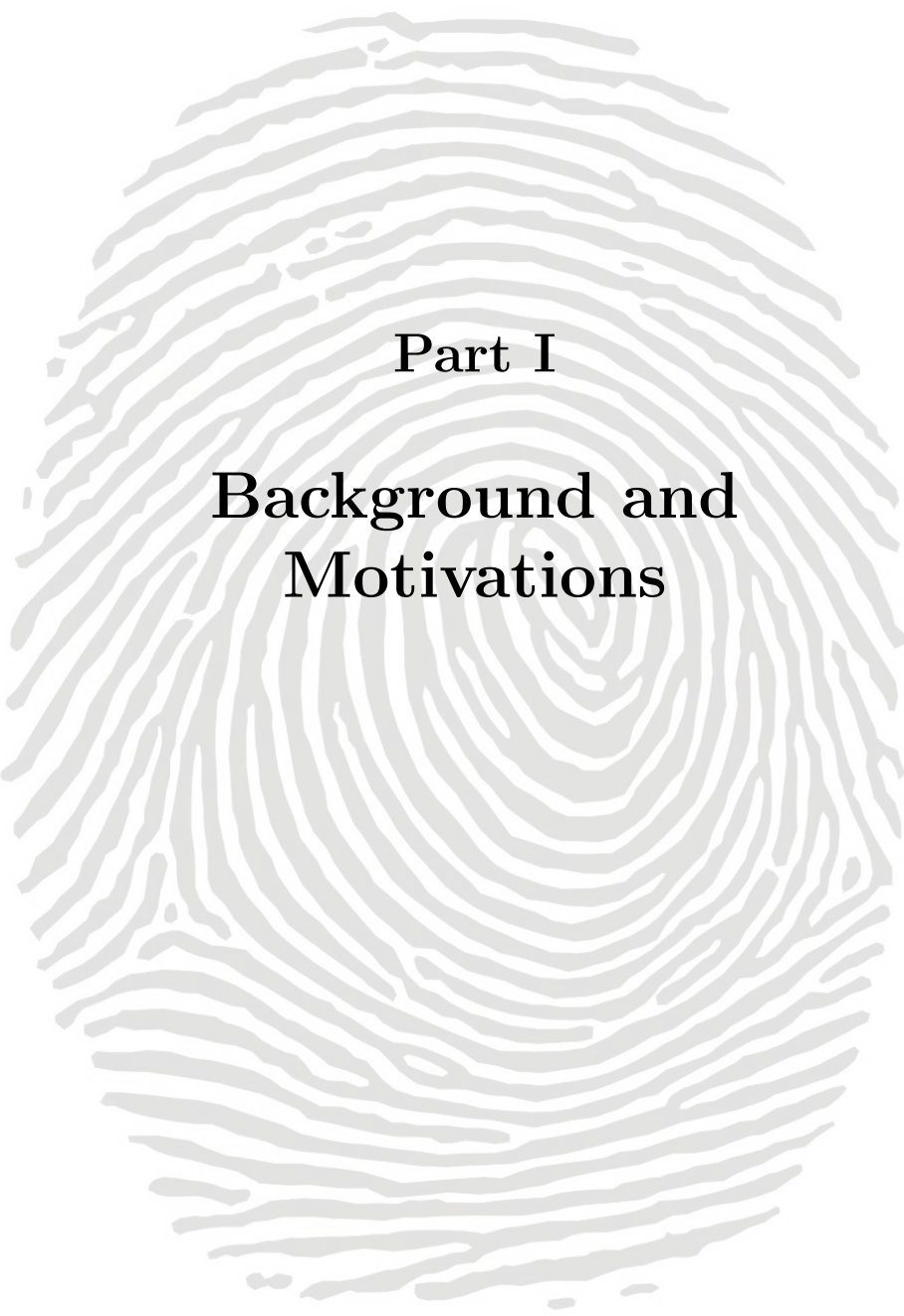
# List of Figures

5.1	a) The <i>interactive</i> rocks are spread on the table. b) The wooden box contains the micro-controller and cables. . . . .	34
5.2	a) The transportable wooden box. b) The table, devised for permanent exhibitions. . . . .	35
5.3	a) The histogram shows the timing of a conflict resolution. b) Assembling the final installation. . . . .	37
6.1	The interactive tabletop . . . . .	40
6.2	Infrared light and shadow tracking . . . . .	41
6.3	Filter pipeline of IR light blobs . . . . .	42
6.4	Filter pipeline of IR light shadows . . . . .	42
6.5	Correction of lens distortion . . . . .	45
6.6	Setup of the <i>t-Frame</i> system . . . . .	49
6.7	These figures show the <i>intra-frame</i> calibration. Some filters are applied to the original image (b) and the horizon is automatically calculated using the standard Hough transform. . . . .	49
6.8	a) Finger triangulation when using multiple cameras. b) The cameras' position is calculated in the <i>inter-frame</i> calibration step. c) An interactive wall application. . . . .	50
6.9	a) When fingers touch the surface on the same area, many intersections are generated. b) This effect produces a lot of false contact points. c) A clustering algorithm group fingers, associating each one of them to the correct hand. . . . .	51
6.10	a) When fingers open, the hand is correctly recognized. b) When the finger are closed che algorithm triggers the event <i>hand-to-finger</i> . c) This approach makes a common desktop display a touch-screen. . . . .	52
6.11	a) Using unsynchronized cameras, the estimated finger position is far from the real position. b) A regression spline is used to smooth down the zigzagged trajectory. . . . .	53

7.1	Automatic recognition of the shape of the closed and open hand: the red area is the segmentation of the hand, the white region is the Convex Hull. . . . .	58
7.2	Comparison of the different manipulations performed on multi-touch table (row 2) and free hand manipulation (row 3). . . . .	59
7.3	Schema of the protocol infrastructure. . . . .	62
8.1	a) b) c) The contents are allocated according to user position. . . . .	64
8.2	a) When using standard interfaces, a zoomed image covers another user space. b) Optical sensors can easily detect the objects laying on the table. c) Zooming an image using a <i>tangible</i> magnifying glass in a multi-touch table. . . . .	65
8.3	Sending a content from a mobile phone (a) to a touch-screen (b), the user can continue his/her personal work. c) The keyboard of a mobile phone is used to type characters in the interactive wall. . . . .	66
9.1	a) A swipe gesture allows access to next or previous document. b) Scrolling the list of thumbnails. c) A shuffle gesture randomly changes the order of visual documents. . . . .	73
9.2	MORAVIA: Web based Video Annotation . . . . .	77
10.1	Touchtables and interactive walls are suitable working environment for cooperative and collaborative tasks. . . . .	82
10.2	Defined by a generic shape, convex, concave and even intersecting polygons, a group is useful to simultaneously act on multiple objects. . . . .	83
10.3	Once a group is dropped on the tag object, the user inserts a name and the folder appears like a simple tag or text. . . . .	84
10.4	A visual feedback helps the deleting task. . . . .	85
10.5	The folder container is explored by means of a continue manipulation of zoom in, while an inner content is opened pressing on its image. . . . .	86
10.6	Our interface allows the users to visually discriminate two different workspaces. . . . .	87
11.1	Pair programming and multi-touch . . . . .	93
11.2	The appearance of the user interface . . . . .	94
11.3	Results of exercise 2 (controversial bugs) . . . . .	100
11.4	Results of exercise 3 (careless errors) . . . . .	100
11.5	Results of exercise 4 and 5 (pattern matching) . . . . .	100
11.6	Results of exercise 6 and 7 (algorithm understanding) . . . . .	101
11.7	Comparison of gesture fluency . . . . .	102

11.8	Geodetic sphere scheme. . . . .	106
11.9	a) The multi-touch planetarium application. b) Full-body version of the planetarium application projected on a geodetic sphere. .	107
12.1	a) Structure of the solid image. b) Interpolation of the distance matrix. . . . .	117
12.2	a) Example of the first 3 levels of the quadtree. The brother nodes are surrounded by a thick line. b) The multi-resolution pyramid. . . . .	121
12.3	a) Accessing the correct tile of the depth image, at bottom level. b) Example of the top-left, and bottom-right tiles. . . . .	123
12.4	a) The image is divided into different tiles. b) The 3D coordinates are displayed in a rounded frame. c) Exploration of a solid image using desktop setup. . . . .	125
12.5	By using simple gestures, the user explores the image and takes a measurement in different multi-touch enviroments: a tablet (a) and an interactive wall (b) . . . . .	127
12.6	a) A user explores a highly detailed 3D point cloud by means of simple gestures. b) Even the major of Cagliari was attracted by our interactive installation. . . . .	129
12.7	a) Panning in the 3D space by means of a single hand movement. b) Performing on-air dual handed gesture, the user can rotate the 3D scene. c) The volume calculation is supported by moving away the hands. . . . .	133
12.8	Distancing theirs hands, users can bring the water level from low (a) to high (b). . . . .	134
A.1	a) A simple algorithm filter the audio signal coming from the bass, determining the raw frequency and the played note. . . .	154
A.2	a) Players execute E major and the lamp lights up in green. When they move to a A major tonality the lamp change its color, showing a bright blue. . . . .	155





Part I

**Background and  
Motivations**



# Chapter 1

## Introduction

When analyzing the evolution and history of human-computer interaction we can see a consistent progression up to the designing of the personal computer, in the early 70's, when mouse, keyboard, icons, menus, windows and metaphors became the columns of interaction.

All of these characteristics have been handed down generation after generation to today's computers. Should we want to be hypercritical, we can say that personal computers have not changed much since they were first designed. With regards to the devices nothing has changed at all: mouse, keyboard, monitor, all of these had already been studied and analyzed in the 60's. At the most, in the meanwhile, keyboard and mouse became more ergonomic and precise and monitors have grown bigger in size and resolution.

As far as the interfaces are concerned, they have certainly evolved in terms of computer computational and graphical capability, which resulted in the visualization of multimedia contents, maps and proper 3D environments. But precisely because contents have changed, since computers have become so powerful to allow visualization of 3D scenarios or augmented reality, the use of mouse, track pad, keyboard and button panels is not longer satisfactory. Changing part of the interaction that include the output and visualization supports, is no longer sufficient, but other components like inputs flow and feedback need to adapt in order to deliver a more interesting exploration.

The actual insufficiency of classic interaction is even more enhanced if compared to interaction of real people in the real world. Let's think about the number of movements we make during the day. Let's think about non-verbal communication, and the countless gestures in the man-to-man interaction. Let's also think about the interaction man-physical object, and the quantity

of manipulations we use to flip through a magazine or a photo album for instance.

There's no doubt that some researchers tried to flip the scenario, by studying alternative interaction paradigms. Visionaries like Bill Buxton were the ones that built the first multi touch surfaces prototypes as far back as 1984, the same year of the first Macintosh release. We can also track back the first kinetic interfaces in the mid 90's based on the automatic hand gestures recognition, hence systems that can recognize a hand or arm movement leaving the user free of having to wear a glove, a wristband, a body suit or hold a pen or a controller.

To this day, computers have been designed to be passive; they do not react to our presence or to our manipulations or gestures.

Given that people use their senses to explore the real world, we must do our best to make the computer exploit our senses and our manipulative and gesture abilities. One of the most recent HCI objectives is exactly this, the study of supports that facilitate the interaction in order to make interfaces more natural or even invisible, without necessarily having to learn how to use them.

## 1.1 Scope and Goals

The main purpose of this dissertation is the design of interactive environments based on more gestural paradigms. To this end we analyse three types of interaction: tangible, multi touch, and free-hand, in order to provide the user with a space where he or she can move freely, using gestures, virtual objects and real objects manipulations. A space where people can act in an easier and faster manner, being the environment shaped on the natural human capabilities.

The tangible interaction uses physical objects to control digital information: people handle real objects representing the state of the system. We can emphasize a change of direction between the past and future of user interfaces, as we notice that in tangible interfaces the relationship between digital and physical is inverted. Rather than make digital objects look like real objects as suggested by the metaphors mechanism (desktop, folders), we attempt the insertion of digital attributes in real objects.

Interactive touch screen displays keep on getting attention because they let users interact directly with virtual objects; thanks to the multi touch functionality and the devices' capability of quickly managing graphic contents, they mimic more convincingly the physical objects manipulation, compared to conventional input peripherals. Although commercial devices are widely available in the market, the *hand made* approach offers a valid method of low cost production



of prototypes that are big enough to host more than one person at the time.

The free-hand gesture interaction gives us the chance to interface the machine and interact in the most congenial way. The ability to intercept human actors movements is quite relevant in the definition of a gesture-based interactive environment. Recent developments on optical technology are making it easy to capture the human body schematic description, like the topological skeleton or the segmentation of significant body parts (limbs, torso or head) allowing the body movement tracking without having to wear or handle any controller.

In the end, the design of an interactive environment should have the objective of expanding the interaction from display to surrounding environment trying to remove the traditional barriers first and foremost between human and machine, and then human-to-human barriers as well, fostering collaboration.

Summarizing, all this leads to the following main objectives of the thesis:

- analyze how tangible, multi-touch and free-hand interaction help the designing of new interactive environments investigating ways in which more natural interaction can be supported;
- tackle the limitations and improve current input sensing technologies to enable their use in limited controlled environments, in particular, settings outside the lab such as museums and exhibitions;
- study how gestures and manipulations can be used in single or combined to design multi-user scenarios based on new interfaces developed with these technological improvements;
- discover a way to evaluate a natural interface in order to measure ‘naturalness’ of developed interactive environments.

Once we will have analysed in depth all these aspects and evaluated our interactive systems, we will apply the lessons learnt to some applications fields in order to ease specific tasks.

## 1.2 Dissertation Structure

The following chapters describe in more detail what we anticipate in this introduction. The remainder of Part 1 goes on to explain the reasons that led us to write this dissertation and shows some applications of the analyzed interactive environments.

Part 2 (chapters 3 and 4) talks about the recent steps of human computer interaction, describing how traditional interfaces evolve into natural interfaces based on gestures and manipulations, clarifying why these interfaces are considered *natural*.

Part 3 (chapters 5, 6, 7 and 8) describes the contributions we give to the state of the art, pointing out the improvements relating to interaction paradigms and the devices sensing performances, so that interfaces can be used even by absolute beginners in a relatively uncontrolled environment.

The dissertation goes on discussing gestural interfaces in Part 4 (chapters 9, 10, 11, 12). Particularly relevant is Chapter 10, describing an interface devised to support social activities and foster collaboration. Chapter 11 describes the performed trials and provides a comparison method testing the different interaction paradigms and sustaining the assumptions introduced before. Chapter 12 reports on some of the gestural interfaces designed for specific application fields.

This dissertation ends in Part 5 (chapters 13 and 14) drawing the conclusions and giving some final comments on the proposed approaches.

Where relevant, the end of each chapter contains a summary highlighting what the focus is.

## Chapter 2

# Motivations

In HCI literature, it is extensively demonstrated that the performance of interactive systems, in terms of being *natural*, is linked to what human abilities are exploited when operating the computer. From this perspective, abilities like object manipulation and gestures use and recognition appear in the early stages of our lives and thus, a gestural paradigm can be considered more effective than common paradigms based on linguistic, visual memory and spatial organization abilities, such as WIMP user interfaces. Furthermore, the exploitation of spontaneous natural abilities, including manipulative and social ones, helps the underlining of the inextricable relation that binds perception, cognition and motor action. Such relation is often referred to as *embodiment*, and is one of the topics of this work.

This way, we decided to focus our studies on tangible, multi-touch e free-hand interaction since they are based on gestures and manipulations (as we will see in next chapters) and the tools of such new interaction paradigm are often considered natural interfaces. In other words, the natural interfaces that are the object of this dissertation can facilitate people's activities and lives, providing an environment where the user can better express his/her own abilities of manipulative exploration and gesture communication.

The tangible interfaces can help a PC illiterate, an elderly person or a child, in their interaction with the digital world through the manipulation of objects distributed in the real world. In the education field for instance, a simple dice equipped with simple inertial sensors could help a person approaching the world of numbers, taking into account the number of dice throwings and supporting the user in calculating simple operations.

Tangible and multi-touch interaction share some of their traits in terms of

direct manipulation. If properly designed multi-touch interfaces allow the use of both hands, exploiting gestures and manipulations in order to explore the contents. Moreover, we do not underestimate other important features. One of the main advantages of multi-touch tables and walls over desktops is that they can host simultaneously a certain number of users and can therefore allow a kind of collaborative or team work.

Free-hand interaction is particularly useful in the application where spatial gestures and full body movements can facilitate and speed up certain tasks. For example, since the human body is used to move around in a real 3D space, it is possible to create interfaces that exploit body parts movements to explore models and 3D scenes. Furthermore a number of applications could use the touch less technology. Free-hand interfaces could be employed to explore medical documents, like x-ray and CAT scans, visualized with the use of hospital displays hence keeping hands free and clean.

Furthermore, in gestural environments, including those based on multi-touch and free-hand interaction, people feel encouraged to exploit their own manipulative and gestural ability. As we intend to demonstrate, this aspect is important because non-verbal communication has a positive impact on many cognitive processes.

Summarizing all the above we aim at the development of interactive spaces capable being of support to a range of applicative fields including education, tourism, sports, rehab, medicine, sociology, 3D interaction and more. On the other hand, we have in mind the wish of creating well-designed and evaluated interfaces, actually developed to make certain tasks easier and to solve very specific problems, without proposing generic solutions.

These and other reasons lead us to the exploitation of manipulative and gestural paradigms in the Architecture and Construction area, as described in the last part of this thesis (Chapter 12). We take such a choice because this area needs alternative solutions, more innovative than standard schemes, that would allow the natural exploration of tridimensional contents to facilitate engineers and technicians work.



**Part II**  
**Related Research**



## Chapter 3

# Towards Natural Interfaces

We begin our dissertation with the analysis of how interfaces evolved from the general purpose of early computers, describing the various phases of this constant evolution.

### 3.1 Interfaces evolution

In the last 60 years, the interaction between human and computer, changed some interface paradigms. You'll notice how each particular peripheral introduces a peculiar communication method. In the end we will summarize on how the most recent interfaces are moving in the natural interaction direction.

**1945: The Bush's Memex.** Vannevar Bush designed Memex (MEMory Extender) even before the beginning of the digital era, it was in fact conceived in the 30's and never developed. Memex worked as analog data archive, considered today's personal computers and hypertext ancestor. Based on microfilms electro-optic technology it was described as an office desk equipped with translucent screens, a keyboard, a set of command keys and levers.

Memex was a system in which anyone could have stored his own books, his own archive and his own personal communication, featuring an exceptional speed and versatility that thanks to voice recognition could be considered as an extension of one's memory. Its most revolutionary trait is the use that Bush intended for it, strictly private and personal. With the extended use of Memex it could have been possible to trace new trails and associative paths in the massive amount of existing information, as to create links among the data.

**1962: The SketchPad's Sutherland.** Inspired by Bush' essay, Ivan Sutherland delivers probably the first interactive graphic interface, when he invented the Sketch-Pad. This system allowed the manipulation of graphic objects by the use of an optical pen: could create graphic elements and move them, change attributes and get a feedback. The system used the first graphic interface ever shown on a display (CRT) capable of plotting the objects by using a system of x-y grid. It is considered as the ancestor of CAD (Computer-Aided Design) modern systems. This study, pretty soon would bring great changes in the way we interact with computers.

**1963: The Engelbart's Mouse.** The first mouse to be used with a computer was designed in 1963 by the inventor Douglas Engelbart at Stanford University Research Labs (Stanford Research Institute) to replace the optical pens used by Sutherland with his SketchPad. It was devised with two wheels placed at a 90 degree angle. By dragging the device on an horizontal surface, the wheels were free to turn hence generating an electric signal that once decoded would give out accurate information on the movements along the x-y axis. The mouse would soon become very popular since it was used as a practical input instrument at the Xerox Palo Alto Research Center, Xerox PARC, where the Xerox Alto was also developed and was using the mouse as the instrument interacting with its rudimental graphic interface.

**1967: Early GUIs, the Engelbar's NLS.** NLS is the contraction of oN-Line System, and it is to be considered as the real ancestor of today's personal computers. This moment marks the date CSCW (Computer Supported Cooperative Work) and interactive computing are born. It was equipped with info sharing mechanism, keyboard and mouse, a windows interface, control system and remote screen sharing. Thanks to the innovative features it was possible to check information by surfing links in a way similar to today's hypertext. Doug Engelbart introduced his creation to the public at The Mother of all Demos, an event that took place in San Francisco in 1968, at the Fall Joint Computer Conference.

**1969: DARPA's ARPAnet.** ARPANET (acronym of "Advanced Research Projects Agency NETwork", also spelt ARPAnet or Arpanet, was conceived and developed in 1969 by DARPA, the USA Ministry of Defence Agency responsible for the development of new technologies for military use. Devised for American military purpose during the cold war, it would become, paradoxically one of the biggest civil projects: internet, a global network that would connect the whole planet. In 1969, Douglas Engelbart and his team at the Stanford Research Institute took part in the first tests hence becoming the second network node.



**1970: PARC's Xerox Alto.** In 1970 Palo Alto Xerox Labs gave birth well ahead of its time, to an experimental project that would be produced two years later for the first time, in 1972: the Xerox Alto. Developed by Palo Alto Research Center (PARC), it's the first computer in history to be fitted with overlapping windows bitmap display and connected to the first laser printer, connected to the first Ethernet network in local area (LAN) and using Smalltalk, the first object-oriented programming language.

**1980s: Metaphors.** When studying the evolution of the user interface we frequently come across a number of metaphors that were used from time to time to create new technological artefacts: the menu metaphor (“the display is a menu”), the desktop metaphor (“the display is a desktop”), the drawing metaphor (“the display is a drawing sheet”), the control panel metaphor (“the display is a control panel”), the room metaphor (“the display is a room”), the agents metaphor (“applications are agent”), viruses metaphor (“virus apps infect genuine apps”).

In fact the metaphoric procedure can be highly stimulating in terms of creativity when designing new software. With regards to the definition of new interaction paradigms, the metaphor suggests new ways of helping the user to build up a new “conceptual model”, or else a mental representation of how to interact with the system.

**1981: WIMP-Paradigm.** In 1981 the WIMP (Windows, Icons, Menu, Pointing device) paradigm is born. It would become the shell of today's user interface and eventually introduced permanently the desktop metaphor and also the possibility of simultaneous multitasking.

In order to control the system, the user had to manipulate graphic elements on the screen instead of entering commands, as it was customary in the traditional systems of those days. With the Xerox Star the first command, What You See Is What You Get (WYSIWYG), were born. In addition to the windows already introduced with Xerox Alto, the concept of icon, a drawing of a corresponding item was slowly making its way: the icon could be opened to interact with whatever it represented, documents or images, email, accessories, calendars or peripherals.

**1984: GUIs, the age of point and click interfaces.** In 1984 the Xerox PARC team gave Steve Jobs the code for Xerox Star graphic interface. Thanks to the WIMP interface, Steve Jobs could give the new Macintosh computers an easy to use and complete interface.

The use of the mouse naturally suggests the idea of showing on the screen “virtual keys” that could be activated by pressing the (real) keys on the mouse. In fact the concept of button, that can be of different shapes or be invisible as in the case of touch sensitive images, has become very popular in this type of

interfaces.

A new manipulation paradigm takes shape whose interaction with the computer is very simplified, and eventually reduced to a simple pushing of buttons pointed by the mouse: point and click. The generalized use of the button concept suggests the natural evolution from desktop metaphor to control panel metaphor.

According to this metaphor, the display tends to look like the control panel of any electronic appliance, with buttons, switches, sliders, warning lights and numerical displays.

**1994: Web Browsing.** The concepts of hypertext conceived by Bush, implemented by Engelbart in his NLS, were later reused by Tim Berners-Lee enriched with the power of graphic interfaces. Such interfaces will largely use menus, various types of buttons, icons and windows. All this will contribute to the expansion of the internet in the mid 90's.

The “surfing” task encompasses an interaction reduced to a set of two elementary actions: point the text or graphic object on the screen with cursor and select it by clicking with one of the buttons on the mouse.

Main element of this interaction through navigation paradigm, The Back Button, would be of great help allowing the user to backtrack the chain of links. It's considered by the experts as the best GUI function ever since the 80's.

## 3.2 HCI revolution

Thinking about HCI future development, we can't foresee with certainty the future trends, but we can list today's most relevant trends that will presumably, influence significantly the interaction design in the near future.

First of all let's think about multimedia. Out of all technologies that served the human race up to ten years ago, the computer processed alphanumerical data, text and simple graphics. The introduction of broadband connection, the large processing and archiving capability of today's systems, allow the user to handle multimedia data, like music and video that could not be managed by systems and networks used in the past. An emerging metaphor is the agent metaphor: the personal computer will increasingly become our personal assistant, obeying our orders and checking these system's large amount of resources.

Research on voice communication has been very active for many decades and resulted in speech synthesis technologies (text-to-speech) and a good level of voice recognition now available in any mobile device. We can therefore assume that in the next few years the main paradigm in man-to-man communication, will be also widely applied to man-to-machine communication.

Development in the virtual reality field will take the user to be fully immersed into an artificial world, with new sensory feedback i.e. the possibility of manipulating virtual objects. The virtual reality applications are quite limited at the

moment, but expectations are quite fascinating: telepresence (the possibility of moving in a faraway environment, seeing what a video camera would see if fitted on a distant robot), tele-manipulation (i.e. sending commands to the limb of a distant robot), computer art etc.

If Grudin [54] described the transition from physical engine to interface and from interface to social space, (other forms of) natural interactions via natural interfaces, like eye and body tracking, will make the interface ubiquitous. The interactive elements we send the inputs with, won't be defined in one object but will be all around us (just like in the case of tangible interfaces) or maybe close to our bodies (think about vital sensors applied to a patient under medical observation). We can already find applications of these concepts to games and amusement appliances: a good example is represented by 'Project Natal' developed by Microsoft for the Xbox360 Console [112]: a depth camera, Kinect capable of scanning via structured light technology, player's position and movement, giving an alternative to the physical controllers used until now.

If we want to transform the meaning of interface used til now, we can say that the new interaction paradigms bring in a new controller: the human body. By the use of body tracking, voice recognition and player appearance, the system will be able to respond to all actions performed and deal with various situation as if it was a real person. It will intercept a certain concern on our face, happiness in our voice tone, or whether we are staring into space, in order to create an experience as real as possible.

We mustn't forget the introduction of medical or psychological applications used to treat patients with social interaction problems. Computers can also help in sports assisting us in training activities, correcting our posture and giving us precious advice while we work out. From primordial GUI to multi-touch, from speech recognition to gestures: the interfaces evolution is very promising.



## Chapter 4

# Manipulative and Gestural Interfaces

In the previous chapter we gave a general overview on the man-machine interaction and user's interface evolution. In describing the latest achievements we used words like natural and direct interaction. The purpose of this chapter is to define these terms, what is peculiar to natural interfaces and the interactions paradigms based on natural gestures and handling. We will eventually choose, amongst the various natural interfaces, a selection of them capable of activating a direct operation. As we will explain in the following chapter, on this selection we will base our work in the attempt to build an interactive environment capable of adapting to general sensory perceptions, cognitive, communicative and expressive human abilities.

### 4.1 What is Natural

The expression "Natural User Interface" is now commonly used. Even some educational magazines talk about Natural User Interfaces and some software and hardware manufacturers have already included Natural User Interfaces in their range of products. Research has been focusing on multi-touch technology and other input methods since 1980's, however we do not have a specific guide yet on design and development of Natural User Interfaces. We can roughly talk about natural interface, NUI Natural User Interfaces to refer to a user interface that is completely invisible to his users or it becomes invisible by a series of interactions.

Generally speaking we tend to accompany the word 'natural' with the word 'intuitive'. This adjective is nevertheless vague, ambiguous, it does not clarify the concept and it does not give a technical description. We need, on the

contrary, to find some boundaries to establish what it is and it is not natural. Although the studies covering this subject are quite recent, there are various definitions of natural interfaces, all of them in line with the definition of a particular natural interface when this exploits inborn or pre-learned skills acquired through experience [86]. The implication is that NUIs will exploit a different skill-set than existing interfaces.

### 4.1.1 Human Skills

In the first place it is important to establish what abilities are needed to use the interface various components and then compare them and find a classification based on pre-acquired abilities and learning abilities. In other words, we're talking about natural interface if the interface works through acquired abilities and we need to analyze at what stage in the human growth cycle these abilities were acquired.

In their research, various researchers [46] considered the characteristics of existing GUIs, comparing the interaction style offered by WIMP to gestural and manipulative interfaces. Considering human abilities, since the skills required to understand menus and icons [62, 122] appear in a later time respect the stage in which we acquire the object manipulation and the use and recognition of gestures [25, 52, 73, 136], the interfaces based on gestures and manipulations are more natural if compared to WIMP interfaces.

In fact, performing a simple task via WIMP interfaces may demand a high cognitive level that is hardly bearable for a user with limited experience. In some cases it is possible to compensate with a period of training, especially in the technology field, as in CAD and 3D interaction. Working in this way the WIMP interfaces diminish the man's capability of limiting the cognitive load under a certain threshold. It is hence necessary to face this problem from a different viewpoint through designing alternative interfaces capable of better productivity.

### 4.1.2 A New Metaphor

With this in mind, George and Blake propose a new metaphor for Natural User Interfaces: Objects, Containers, Gestures, and Manipulations (OCGM) [46]. It's easy to understand how the concept of OCGM refers to a metaphor: objects are metaphors for units of content or data. Containers are metaphors for the relationships between contents. Gestures are metaphors for discrete and indirect interaction, and manipulations are metaphors for continuous, direct, environmental interaction.

### 4.1.3 Gestures and Manipulations

To shed some light and revise the meaning of gestures and manipulations, we can say that a gesture is a body movement intended to communicate or interact with the surrounding environment [84,105]. On the contrary, manipulations are more linked to the physical meaning, we handle and move an object with the very intention of changing its position, put it farther or closer maybe to better inspect it. This distinction leads to considering a gesture like a discrete action, with well defined beginning and ending moments distributed over the time and we can recognize the gesture only when the action is completed. Are defined as gestures, the hand movement that says hallo or goodbye, the double click on a folder and so on. On the contrary actions like moving a physical object or dragging an icon on the screen, are considered continuous actions.

### 4.1.4 More Natural Interfaces

Some aspects of human interaction have been studied with the objective of establishing what skills are used with WIMP and OCGM. Specifically, different studies of early childhood cognitive development [46,106] demonstrate that interfaces using OCGM require minimal cognitive loading, cognitive skills developed very early, so they are intrinsic and come naturally and use skills-based behaviors.

The OCGM paradigm provides a pattern that allows the creation of more natural interfaces. Moreover, researchers [46,70] recommend concrete metaphors design that use skill-based behaviours and Reality Based Interactions (RBI). These approaches moved interfaces closer to real world interaction by increasing the realism of interface component and letting the users interact with them in a even more natural way, as people would do in the real world.

## 4.2 Chosen Interfaces

As seen in the previous section the OCGM paradigm offers strong points of more natural interfaces design. These interfaces via RBI scheme take the human-computer interaction closer to the human-human and human-real world interaction.

Amongst various types of RBI styles, this Ph.D. dissertation analyzes the interfaces based on gestures and manipulations, in particular tangible, multi-touch and free-hand interfaces. Much of the effort is geared towards improving the devices and the interactions paradigms so that the interfaces could lighten up the cognitive load during interaction, allowing the user to exploit either innate and learnt skills when executing a certain task.

Before going into detail on tangible, multi-touch and free-hand interfaces, let's

have a overview about gestural interfaces in HCI, underlining the cognitive and communicative role of gestures and manipulations.

### 4.3 Gestural Interaction

Gestures in Human Computer Interaction have been variously addressed by computer scientists. Starting as part of multi-modal user interfaces (following seminal work by Bolt, [8]), gestures have recently become a dominant aspect of tangible interaction [68], kinetic interaction [17], emotions recognition [26].

However, research in gestural interaction so far has concentrated rather on sensing gestures, (e.g., Pavlovich and colleagues [105]; Wu and Huang [137]) than on defining what a gesture is and how to give meaning to gestures in interactive systems often tailoring this issue to the specific needs of applications (e.g., pen computing) or technologies (e.g., multi-touch displays). Additionally, gesture has been mostly regarded to as an alternative, rather than a companion, to other input devices, that allows a more natural form of interaction.

On the other hand, gestures and non-verbal communication, together with those human activities and social interactions which it is functional to, has been the subject of deeper investigation by anthropologists and psychologists.

Among the many movements that we perform with our body, gestures are, according to Kendon, intentional excursions that are meant to convey some message [77] and can be classified along a continuum that spans from gesticulation to sign languages, passing through speech linked gestures, emblems and pantomimes. McNeill underlines how such categories are in relation with the presence/absence of speech and linguistic properties; gesticulation is (almost) always accompanied by speech, emblems may or may not, while sign languages (almost) never are. Additionally gesticulation doesn't have any linguistic property, emblems have some linguistic properties, sign languages have full linguistic properties [94].

This cognitive role of gestures has an analogous in manipulations. Manipulations are actions performed on objects in order to change the state of the world. Their potential is being explored in the context of tangible interaction, and the role that manipulations play in human cognition has been explained by Kirsh and Maglio [80] in terms of epistemic and pragmatic action, the former being actions performed in order to improve cognition, where the latter are planned and performed to reach a specific goal.

The advantages of such behavior are that: (i) the complexity of the task is moved from the head of the user to the world, available strategies and possible solutions to a given problem appear at a glance; (ii) the (limited) resources of attention and memory are not wasted to concentrate on the strategy and can be used to explore alternative solutions; (iii) such exploration performed by means of manipulations on the world (or tools) are easier (i.e., they require less



cognitive effort) and faster (i.e., they require less time) than it is to do the same mentally.

There is also strong evidence that epistemic action increases with skill [91]. This means that in HCI this is not a behavior of naive computer users, but rather a powerful feature to leverage in interaction design. Tangible and multi-touch interaction, both build on this concept of providing direct manipulation of physical/graphical objects.

But interfaces don't define the interaction. Users' behavior is far more rich of what can be recognized and supported by the interface. In [51] Goldin-Meadow and co-workers show that gesturing lightens cognitive load while a person is thinking and explaining how she solved a math problem, resulting in an improved performance in a short term memory exercise. Similarly, Cook and coworkers [31] show how gesturing during a learning session helps children retain the knowledge gained.

Nonverbal communication in a group is also a positive behavior indicating healthy cooperation. Morrison and colleagues [97] show how the introduction of an electronic patient record in a hospital can disrupt some virtuous practices, partially voiding the benefits of the digital support.

## 4.4 Tangible User Interfaces

Tangible interaction attempt is to fill up the gap between physical and digital world designing objects and environments in which people can control the interfaces in a more physical way. Tangible User Interfaces (TUIs) allow the exploitation of different natural abilities, such as manipulative and social ones, focusing on the inextricable relation, often referred to as *embodiment*, that binds perception, cognition and motor action.

### 4.4.1 Physical Context

The Ubiquitous Computing concept was first introduced at the beginning of the 90's foreseeing the use of physical artefacts to represent and control digital information. Various researchers experimented the consolidation of electronic and physical world creating the first tangible prototypes: Wellner conceived Digital Desk [128], an interactive desk on top of which digital contents can be physically manipulated; Fitzmaurice et al. designed Bricks [43], bricks that function as physical controls for electronic contents; Ishii e Ullmer came up with Tangible Bits [68], an innovative paradigm of physical interaction to grasp and manipulate digital information.

Based on these prototypes, a new interaction concept comes to our attention, a "tangible" interaction, that incorporates the notion of interaction focused on visualization and subsequent elaboration of information, with a notion centered

on action. Such a deep integration strongly grounded in the physical world of representation and controls, draws a fundamental divide from the concept of graphic interface peculiar to HCI <sup>1</sup>.

## 4.4.2 Actions and Representations

In tangible interfaces, users manipulate physical objects that are distributed in their surrounding environment. However, even in standard interfaces people use physical objects, such as the mouse, the keyboard or the computer display. In traditional interfaces we send a command through input actions and we observe the effect in the output peripherals. Whereas in tangible interfaces we have a coincidence of the spot where the action takes place and the place where the effect is produced.

A cognitive process takes place during the interaction. The cognitive system provides resources can be represented internally in the individual's mind or externally via artifacts that can be represented with various forms. For these reasons, the nature of representations and their distribution within internal and external resources is at the base of tangible interfaces design, and many researches are of great help in understanding how to properly design these environments.

In the end, we can summarize the characteristics of tangible interfaces with two simple observations. First, the user acts on real objects themselves, not a digital representation. Second, we can say that the user interface is not meant to represent the state of the system, but rather the interface is the state of the system.

## 4.4.3 Tangible Interface Design

The design of interfaces can be influenced by implications due to the fact that some parts of interface might be moved outside the screen. In fact we can build objects and physical spaces equipped with digital properties. This led designers to explore new techniques, pushing our physical abilities. With a proper interface, for example, both hands can be used simultaneously, being able to achieve our objectives more naturally.

**Manipulation** With tangible interfaces people can obviously exploit their ability of manipulating objects. We can use materials with tactile qualities that can be physically manipulated. This provides a more physical experience when using machinery, touching the emotional and sensorial sensibility.

---

<sup>1</sup>P. Marti, "Theory and Practice of Tangible Interaction" (S.A. Iacolina translation of "Teoria e Pratica dell'Interazione Tangibile"), pg. 337 [118]

**Space** Using tangible interfaces people act in the real world, moving and manipulating objects spread in the space. The perception of the space is extremely important. This can be easily highlighted by thinking how we act in the everyday life. People are dynamic, we move constantly, going around in the space that surrounds us, even when we are just thinking. We can say that people stay put, sit on a chair, only when using a computer. Hence tangible interfaces make us move and use our body, and our senses to take action.

**Social** Relationships between physical objects that surround us and analysis of their particular configuration help the comprehension of the social and cognitive processes. Logical and physical space is organized in such a way to suggest specific interaction modalities and opportunities. At same time the object specific physical representation urges to communicate and interact, making its use irresistible. Paul Dourish [37] claims that tangible interfaces through embodiment allow the exploitation of the easy use we make of physical artefacts in order to mediate communication facilitating social exchange.

## 4.5 Multitouch Interfaces

The evolution of multi-touch interaction is strictly connected to development of devices that permit such interaction. This type of sensing technique is not a new concept, as it has been around since the 70s'. As reported in Schöning technical report [115], touch surfaces that use optical systems through sensors and video cameras have been patented [45, 76, 92, 99, 129] and widely accepted. The industrial development of resistive surfaces allowed the production of desktops equipped with single touch screens. The microfilm viewer systems, equipped with joysticks and push buttons become obsolete. The peripherals with keyboards and button panels are replaced with others that are touch sensitive. It is now possible to place interactive information points in communal areas, for the user's easy access to information. Capacitive surfaces will follow soon and finally through multi-touch screens the user is able to use both hands [Buxton:1986nj,Hinckley:1998:TVM:292834.292849] in a series of different fields of application. The innate and immediate interaction offered by these innovative multi-touch devices, ensured that an ever growing number of people familiarized with them. As the years go by, touch technology becomes increasingly used in the mobile devices industry, whether it is notebooks, tablets or smart phones. Interfaces become more fluid, dynamic, adapting to human beings needs and handling capabilities.

At the beginning of 2000 researchers found out alternative techniques for the development of multi-touch screens. The re-appreciation of the Frustrated Total Internal Reflection (FTIR) *precept*, started by Han's work [55, 56], and other innovative optical technologies (like Diffuse Illumination [115]), introduced newer and cheaper techniques for the development of optical multi-touch systems.

Thanks to the availability of information through internet and also of the necessary components, we enter the low cost peripherals age. The simplified production process and use of these devices compelled the big brands to become interested in large dimensions multi-touch surfaces production, particularly fit for a social and co-operative context.

Today's multi-touch systems are reactive and responsive, capable of recognising gestures and manipulations. In his personal web page [21], Bill Buxton analyzes the different shapes of multi-touch devices and various interaction principles. Given that some of his consideration helped us with the selection of appropriate interfaces, we report herewith a short list of and some of our points of view.

**Gesture vs Manipulations:** we can distinguish two types of interaction also in multi-touch interfaces. We talk about direct manipulations or simply manipulations, which can also be regarded as online gestures, when the user activates a continuous action to alter some of the interface properties, like scaling and rotating. In contrast, offline gestures, simply called gestures, are usually processed after the interaction is finished; i.e. a circle is drawn to activate a menu, two fingers are joint together, performing a finger-hold one finger pressed for a certain amount of time, or a double-touch. In these examples the associated command is performed only once the interaction is completed and the gesture has been recognized.

**Discrete Actions and Continuous Actions:** when using traditional multi-touch interfaces we use a large set of discrete actions, opening folders with a finger touch, like they were a "push button", or a QWERTY physical keyboard. This it have inherited since the time when people used to push a button and enter commands. But in the real world, we always use continuous actions in every day life, like opening a drawer or a window. Since nowadays we can use devices capable of reacting even to manipulations, couldn't we develop interfaces based on our abilities? We will further investigate this aspect in section 10.

**Single-finger and multi-finger:** even if the original studies on multi-touch devices date back to the early 80', the majority of touch surfaces only permit single-touch interaction. Single point sensitive interfaces, regardless of the peripheral used (mouse, touch screen, joystick, trackball), are very limited. Since we have multiple fingers it is high time technologies adapted to our abilities.

**Multi-point and multi-touch:** Buxton says that many, if not the majority of so called multi-touch are in reality multi-point: just because we use a track pad instead of a mouse, we don't consider a new technique interactive when working with a portable computer. Double click, dragging, drop down menu identify the same interaction regardless of the fact that we use a touch pad, a trackball, a mouse, a touch screen or a joystick.

**Multi-hand and multi-finger:** some input paradigm support the simultaneous use of different fingers in the same device, some other require the use of separated devices. Selecting the appropriate system is important: the development of interfaces allowing the use of both hands is not enough, we also should try to use our hands on the same device.

**Multi-person and multi-touch:** when more people use simultaneously multi-touch interfaces, some peripherals can discriminate the users, linking the different finger pressing to the user doing that action. With proper technology [36] it could be handy the development of interfaces that can discern different work areas, one for each user, on the same screen. We will go into this subject in the next parts of this dissertation (sections 10 and 8).

In chapter 6 we enter in more detail on design and development of low cost multi-touch devices.

## 4.6 Free-Hand Interfaces

In the last few years research has focused on finding new techniques in order to capture the user's movements in the three dimensional surrounding space. Beyond theoretical aspects, a real gesture recognition implementation requires the use of ad-hoc hardware instruments like sensors or tracking devices. As analyzed by Mitra and Acharya [95], those devices can be classified in two categories:

- Wearable tracking devices, gloves, suits, controllers and similar.
- Computer vision-based devices and techniques, video cameras paired with algorithms that find movements from the video.

Wearable tracking devices are very accurate and can reveal sudden movements, like fingers movements while moving hands; on the other hand, methods based on Computer Vision are less invasive and are able to identify also colors and textures.

### 4.6.1 Wearable and graspable tracking devices

We can use tracking techniques to acquire some of the motion's information, like position, speed and acceleration. In general acoustic, inertial, LED, magnetic or reflective markers, or combinations of any of these, are tracked. We will now describe the two main methods in use as an example: magnetic and optical

systems. They summarize the advantages and limitations of various technologies in motion capture.

One of the mainly used techniques exploits the magnetic field properties: a transmitter emits a constant magnetic field while the user wears an overall fitted with sensors and capable of tracking the intensity of the magnetic flux. Both position and orientation of each body part can be accurately calculated by mapping out the relative intensity during motion.

Optical systems utilize data captured from image sensors to triangulate the 3D position of a subject between one or more cameras calibrated to provide overlapping projections. Data acquisition is traditionally implemented using special markers attached to an actor.

Motion capture systems allow the acquisition with high precision even of the limbs rotations, giving up to 6 degrees of freedom per marker. However these systems are used just in filmmaking and virtual animation. Having to wear markers on body suits makes these technologies very expensive and not very practical in the home environment.

From the simplification of these technologies comes the production of devices that are more suitable for the common user. This is especially true in the entertainment field of video games where various devices provide force feedback and are able to sense the user's motion. Hybrid console controllers, equipped with accelerometer and optical sensor technology, like Nintendo Wii Mote, offer the feature of motion sensing capability, which allows the user to interact with and manipulate items on screen via gesture recognition and pointing. These controllers respond to motion and rotation for enhanced control as you swing, swipe, thrust, or turn the controller. The abilities to move and gesticulate into 3D space are exploited in various simulated sport activities, allowing for a good user's performance.

## 4.6.2 Free-hand interaction

Although the domestic systems, like above described controllers, are able to acquire the body's movement with a good precision, these systems have some limitations. These system's most important limitation consists of the necessity of holding a controller in your hand which tends to preclude extreme performance movements. The interaction is mediated and the use of the device is sometimes not realistic, like in running and swimming.

More recent Computer Vision systems are able to generate accurate data by tracking surface features identified dynamically for each particular subject.

Among different gesture recognition methods based on Computer Vision, depth-cameras are often used: an example of use of depth-cameras in gesture recognition is the work of Benko and Wilson [4], where they used the 3DV Systems *ZSense* camera (since June 2009, *ZSense* is part of Microsoft).

Depth cameras capture the *range image* (i.e., the per-pixel distance from the camera to the nearest surface) and have the potential to drastically increase

the input bandwidth between the human and the computer. Such cameras enable inexpensive real time 3D modelling of surface geometry, making some traditionally difficult computer vision problems easier. For example, simple algorithms extract the topological skeleton of a human body from the range image, track user movements along consecutive frames and capture the presence of physical objects.

The last Microsoft entertainment product, *Kinect*, incorporates a depth-camera; this device, commercialized as a game controller, allows users to interact with the console by moving the body, mainly hands and arms, in the real space.

To make this interaction possible, this device embeds several sensors: a regular VGA video-camera, an infrared projector and a sensor that reads the environment response of an infrared light pattern released by the projector. This system is sufficiently accurate since it produces data with three degrees of freedom (3DOF). Latest drivers calculate rotational information from the relative orientation of three or more nodes. A freely available API decodes the raw signal and provides developers a digital description of the human body in 3D, recognizes different body parts (head, neck, shoulders, wrists, hands, hips, knees, ankles, and feet) and therefore can create a digital reconstruction of the human skeleton.

## Manipulations, Gestures and Selection

Obviously, also kinetic and free-hand interfaces support gestures and manipulations. Moreover, we can affirm they provide new and alternative interaction modalities with the virtual world. Such interfaces make the users free of manipulate objects represented on the screen in a way that is very close to the way we operate in the real world. We can, doing a continuous movement, stretch out our arms or else move in the physical space around us to move an interface item or zoom the object shown on a display. The manipulation task becomes therefore simpler and simpler, more fluid, more natural.

To interact with an interface we also need separate events, necessary for the selection task completion. In traditional graphical interfaces, it is possible to select and to press on a button or on a menu item via a mouse click. In other words to action a command it's necessary to recognize a separate event limited in space and time. With this purpose in mind research has focused on analyzing the 3D movements of a topological skeleton node and recognising some predetermined gestures. For example, by positioning the hand on your head or keeping it in a certain position for some time, the user can for instance make mimicking the click of a push button.

The importance of gestures in free-hand interaction is highly demonstrated in literature. For example Francese and colleagues [44] measured the presence and the immersion of different 3D gestures interaction techniques (namely remote-based and full-body). In the end, they concluded that the perceived immersion

of such an interaction technique is high and that the users pass quickly from a novice to an expert style of interaction.

Above all, due to the lack of precision and resolution, at the moment of this PhD thesis depth-cameras and motion capture devices are commercially used in video games applications. We are witnessing the arrival of higher resolution devices although reduced in scale, like the LeapMotion™ [98] and Kinect™ for Windows v2, they open many interesting scenarios most of all because this type of interface presents a touch-less kind of interaction [58, 98, 113, 127].

### 4.6.3 Gesture recognition

Gestures can be represented as multi-dimensional and time-dependent data, so a classic approach for their recognition is the use of Hidden Markov Model (HMM) [109, 110]: this is a valid method to recognize time varying data and consists of a network of nodes, transitions between nodes and transitions probabilities, starting from defined input symbols. These input symbols must be discrete and because gestures are multi-dimensional data it's mandatory to perform a discretization, viable in different manners and used in some works with HMM: manually specified conditions [123]; Self Organizing Map (SOM) [82] used by Iuchi et. al [69].

Other gesture recognition methods use state machines, as proposed by Matsunaga and colleagues [93] that used Support Vector Machines (SVM) [93] for transition conditions learning, while instead Oshita [93] used manually specified fuzzy-based rules. In any case the state machine is created manually and it's not an easy process: this can be an important limitation, but the work of Oshita and Matsunaga [102] obtained important results; they used SOM to divide gestures in phases, thus to create the state machine, and finally they used SVM to determine transition conditions between nodes.

## 4.7 Objectives Updated

The research carried out for this dissertation is heavily influenced by the observations reported above; by now our objectives are:

- create interactive environments where people can use their senses;
- improve various sensing technologies performance with the objective of giving the user the chance of expressing communicative and manipulative abilities at their best/the best they can;
- investigate on gestures and manipulations considering whether some interactions based on gestures could be associated to manipulations instead and vice versa;



- observe how people use the gesture language and create interfaces that ease interaction.
- develop interfaces that support multiple users, discriminating them and allocating a working area to each one of them;
- evaluate the developed interfaces and compare various interaction paradigms.

We will continue this dissertation with investigating and examining natural interaction paradigms trying to follow the objectives listed before.





## Part III

# How to set-up an Interactive Space



## Chapter 5

# Tangible Interaction

People have an incredible ability in perceiving the external world through their senses. That's why the interaction with a computer should be very similar to the interaction with the real world. When using the standard GUIs our innate abilities are not fully exploited and stay dormant. On the other hand tangible interfaces supply a more realistic experience in both the contents comprehension and manipulation aimed at modifying the system state.

For these reasons we decided to explore and study the manipulative interaction starting exactly from tangible interfaces. In this chapter we describe one of our installations, based on an interactive table, that we used to make our early experiments and to gather the feedback from real users about manipulative and tangible interaction. Our intent is to exploit this experiment and tests to find a term of comparison and, in further steps, to build or improve other types of interfaces, including multi-touch and free-hand ones.

### 5.1 Choosing the appropriate application

At the moment of choosing which application to use for the exploration of the manipulative interaction, we were spoilt for choice.

With regard to the development of tangible supports, we free our creativity using materials and devices that are easily available [60, 64, 131]. Analyzing other works [59, 74], we can also say that the making is strictly connected to the type of application we are developing and the consumers being targeted.

As previously described, tangible interfaces allow people to use a real object to interact with computers. They can be adopted to make interaction easier in the design of applications to be used even by less experienced users. This way, tangible interfaces are particularly useful in those applications where physicality is a prevailing trait, like in exhibitions and fairs where we often find displays and information points, or in museums, with applications linked to cultural

heritage and tourism.

After all these considerations, we began to design an interactive application aimed at the Tourism Industry in Sardinia. Our goal is to provide users with an installation that can be used to get information in the most comfortable manner. We devised a tangible interface exploiting three senses, touch, sight and hearing, that offers the visitor a glimpse of Sardinia.

The final idea is to build an interactive table on which some autochthonous rocks and minerals are laid. The moment you pick one of the rocks, the sensors connected to the screen start playing videos and music that narrate the story of the area where the mineral came from, accompanying the visitor in a journey in the architecture, archaeology and culture of Sardinia.



Figure 5.1: a) The *interactive* rocks are spread on the table. b) The wooden box contains the micro-controller and cables.

### 5.1.1 Exploiting our senses

Our objective is to tell the story of specific areas of Sardinia linked to a particular area. The region of Sardinia is historically divided into different areas, each one of them with its own geographical and historical identity. The narration ought to describe the various aspects that typify and differentiate them like architecture, archaeology, folklore, civic, religious and sports events, tourist attractions, handicrafts and geography.

Obviously at the state of the art we have various supports for consulting tourist guides [18, 19]. Audio guides play a recorded excerpt and leave the user free to move around within the exhibition space. Multi touch devices [15] and interactive info points also deliver additional multimedia contents. However from an interactive point of view, these solutions are limited because the control of the user experience is embedded in the same peripheral. The ‘augmented’ book [40, 125] offers a more natural use of the multimedia contents moving it closer to the reading of a book. We wanted to offer a more physical experience than browsing the pages of a book.

We decided then to extract from the digital world an object that the user could touch with his own hands and that would be linked at the same time to the contents to be displayed. We started from the selection of 9 rocks and minerals representing the various areas of Sardinia, each one of them corresponding to a specific area, including for instance sandstone from Cagliari, granite from Ogliastra and sand from Is Arutas (Oristano). Rocks well represent the raw material of the different local architectural and archaeological structures, but on their own can't offer a full description of buildings and landscapes and can't give an insight on other important activities, like sports or handicrafts for instance. With the objective of giving the user a complete experience, we selected 9 video clips with evocative music: each video is linked to a particular stone, hence to a particular area and tells a different story. Music also helps distinguishing the facets of each territory and involves the user in a deeper way. This way, the interactive installation offers the user a suggestive interactive experience through the use of the three senses: touch, sight and hearing.



Figure 5.2: a) The transportable wooden box. b) The table, devised for permanent exhibitions.

## 5.2 Interactive Table

Once we put together multimedia contents and objects we could design the interactive structure. In our installation a table contains a standard LCD monitor and various objects are spread on the table (figure 5.1(a)). The interaction we proposed is very simple. When idle, the display shows static images like a screensaver. When the user picks a rock, the associated video starts playing on the monitor. Visitors can weight a stone, stroke it and see the video in detail and listen to music. When the stone is put down a video transition takes the system back to idle state.

## 5.2.1 Multiple Selections Conflict

Given that the installation has only a normal size single display, the interaction is obviously single-user and can only see one video at the time. Each video is 90' long so that the installation can offer an exhaustive but synthetic description of each area and that the contents display doesn't get stuck. Lifting up an object while holding another one in the user hands makes the video play a message inviting the users to put them down and lift just one at the time.

## 5.2.2 Sensors and Architecture

Thanks to the availability of micro-controllers is now quite easy to go from project to development of apps that make use of sensors, switches and actuators. The MCU characteristics belonging to Arduino family, Raspberry Pi and others are by now standardized: all of them have a number of input and output digital and analog ports and they are powerful enough to be able to perform a handful of calculations in real time. There's a huge community right now, thousands of people using micro-controllers to do thousands of different things. Any hardware peripheral you want to talk to, someone has probably figured out how to connect it to an Arduino, for example.

Our demands were quite simple: get the application to react when one of the objects is put down or lifted up from the table surface. In order to calculate the distance of an object from the table we used light sensors connected to an ArduinoMega analog ports. Once programmed, the micro-controllermicrocontrollers sends in output to the serial port the values retrieved by the sensors (figure 5.1(b)). The nearer the object is to the less the light that hits the sensor and the smaller will be the output value. Each sensor is therefore associated to an object and a video. We also developed an application for a classic computer that after reading the values inputted in the serial port, determines if an object is lifted up or put down and starts playing the corresponding video.

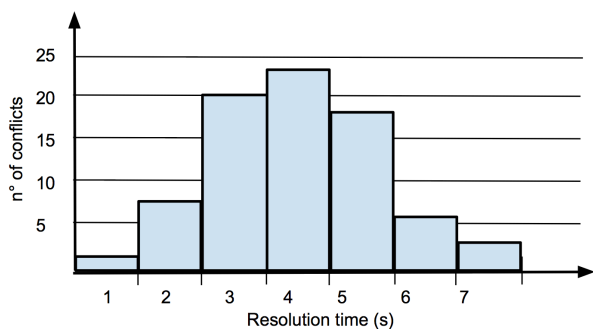
## 5.2.3 Transportable and fix versions

The prototype first version is just a 1,20m x 0,80m x 0,10m wooden box (figure 5.2(a)). The top side is drilled to allow fitting the sensors while the inside contains micro controller and cables. The LCD display and computer are external. This version was devised to simplify the transport and the installation of this appliance.

Considering the ever-growing demand of interactive appliances we fit the sensors on a different structure, devised for a permanent exhibition. The table is 2,20m long, 1,20m wide and 0.12m thick (figure 5.2(b)). The top surface is a 8mm thick sheet of steel, whereas the table core is made of solid wood, containing electronic circuits slots. This prototype includes a monitor and a computer,



looking like as one piece, one structure. Obviously the remarkable weight of 160 Kg and its volume made the installation of components not easy, but the prototype makes it up by showing an excellent design.



(a)



(b)

Figure 5.3: a) The histogram shows the timing of a conflict resolution. b) Assembling the final installation.

### 5.3 Evaluation and Discussion

The interactive appliance has been used in two different real scenarios: a fair in Copenhagen and the Sardinian Store in Berlin. PickARock was installed in its mobile version at the “Sardinien for alle sanser” for 8 days in Copenhagen in October 2013 and in its final version in Berlin in December 2013, where the installation was hosted within the photographic exhibition for the promotion of the Sardinian Store. The appliance was active for 20 days, during which time the visitors tried it out to learn about the Sardinian culture.

During these events, many people used the appliance. The feedback and comments received have been overall positive. What was widely appreciated is the possibility of controlling the appliance through manipulation of the real objects, familiar to the users.

This installation can be considered as an interactive information point where the menu is represented by objects laid on the table. Picking up a rock equals to selecting the desired item to be visualized and once you pick up an object the audio-visual contents begin to play. Not only the selection happens in the real world, but also keeping the object in your hands indicates the state of the system, and plays back the associated video. This way, the user interface essentially becomes invisible and we can interact with the application in a natural way.

The interaction type is single-user. This has not unsettled the user experience. We did not manage to obtain filled out questionnaires, due to the fact there were

too many people visiting. However our application recorded a log, reporting the user activities. As we can see from the histogram in figure 5.3(a), the users automatically avoided and resolved the potential conflicting situations that would be the result of simultaneous lifting of more than one object. This reflects the fact that the lifted stone represents the state of the system.

This kind of installation can also be considered as support for visually impaired people. In fact the users can feel the surface and the weight of the stones, listening to the related music. Something that cannot be done with traditional GUIs.

## 5.4 Summarising

This early experience allows us consider the manipulation of real objects as a mean of human computer communication. The experiments we made proved that the possibility of getting touristic information through manipulation not only eases the interaction, but also encourages and attracts people to use the application. In other words by using touch, hearing, and sight the proposed interface offers a greater involvement in its use; and the touch and feel experienced helps to better fix and evoke the acquired information.

These considerations will prove helpful and will serve as term of comparison when analysing other types of interaction to be discussed at a further stage.

### **Acknowledgement:**

The installation of PickARock was developed in collaboration with CRS4 natural interaction researchers. The original idea was born from the collaboration between NIT of CRS4 and the ‘Centro Servizi promozionali per le imprese’ of the Chamber of commerce of Cagliari within the ‘Sensi di Sardegna’ presentation, a project sponsored by the Chambers of Commerce of Cagliari, Sassari, Nuoro and Oristano.

## Chapter 6

# Multi-touch Sensors Improvement

Nowadays we live in a technological world, pervaded by an ever growing number of digital and multi media contents. We experience the need to support exploration of information and interaction with the devices that provide them. Giving virtual objects a more specific identity, touch screens keep getting our attention. Thanks to them we can better exploit our manipulative and gestural abilities through actions that are very much like manipulations of physical objects like images, maps or documents. As we demonstrated in the previous chapter, the manipulative action eases interaction and let us consider it as one of the keys of natural and usable interfaces. Other important aspects are sharing and co-operating. If a touch screen is big enough we can share it, hence speeding up the pursuing of personal and shared objectives.

With this objective in mind our research focused on the construction of two shared screens, a multi touch table and a touch wall all three of them allowing multiple users to work simultaneously and collaboratively. By adopting a *do-it-yourself* approach, their construction cost doesn't require an excessive investment and the necessary components are easily available in a standard laboratory.

To begin with, we'll describe how we worked on improving the FTIR technique. Our efforts were aimed at the development of a multi-touch platform with enhanced responsiveness to environmental light, hence suitable to be placed in particularly busy open spaces where people are more likely doing team work. We then take our study back to the development of touch-wall. Our work can be considered as the evolution of an existing innovative prototype that exploits the optical technology. Let see how we faced various challenges, discussing the solutions found and what was the conclusions we came to.

## 6.1 Multitouch Table

Multi-touch displays offer a suitable working environment for computer supported co-operative work and foster the exploration of new forms of social computing. Frustrated Total Internal Reflection (FTIR) is a key technology for the design of multi-touch systems. With respect to other solutions, such as Diffused Illumination (DI) and Diffused Surface Illumination (DSI), FTIR based sensors suffer less from ambient IR noise, and are, thus, more robust to variable lighting conditions. However, FTIR does not provide some desirable features, such as finger proximity and tracking quick gestures.

To partly address such issues, in this section we propose to take advantage of natural uncontrolled light, using the shadows projected on the surface by the hands to improve the quality of the tracking system. The proposed solution exploits the natural IR noise to aid tracking, thus turning one of the main issues of MT sensors into a useful quality, making it possible to enhance tracking precision and implement pre-contact feedback.

### 6.1.1 Building a Reliable Sensor

As described in the previous part of this dissertation, multi-touch displays represent an intriguing research field that, recently, has gained new attention.

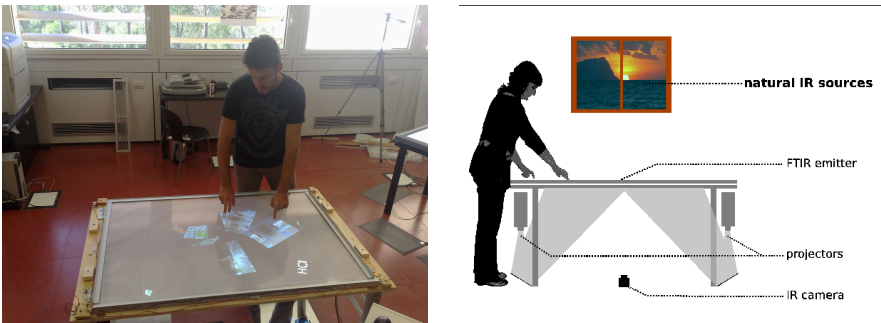


Figure 6.1: The multi-touch interactive tabletop: the picture was captured while using the table (left) and schema representing the overall setup of the table (right). Notice the operational conditions: strong direct lights and sharp variations of the luminosity.

Let's provide here more details about a key technology for the design of low-cost multi-touch systems: Frustrated Total Internal Reflection (FTIR). Common FTIR setups [55, 56] have a transparent acrylic pane with a frame of LEDs around the side injecting infrared light. When the user touches the acrylic, the light escapes and is reflected at the finger's point of contact. The infrared

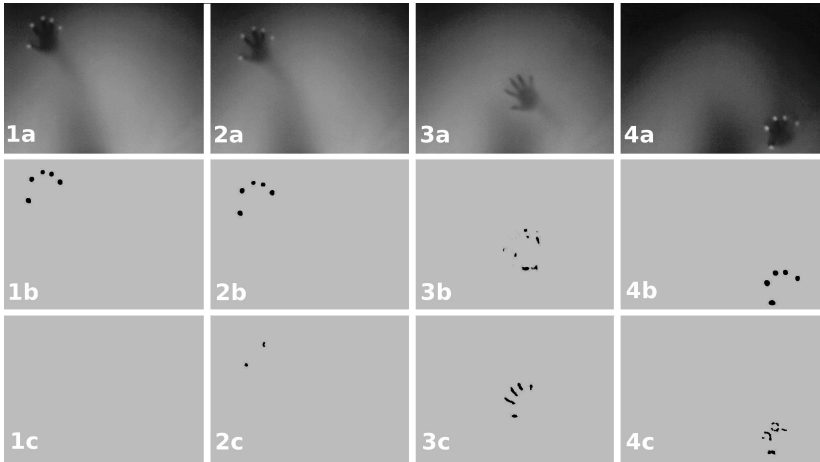


Figure 6.2: The result of tracking on IR light and IR shadow.

sensitive camera at the back of the pane can clearly see these reflections. Being the acrylic transparent, a projector can be located behind the surface (near the camera) yielding a back-projected touch sensitive display. The software framework relies on a set of computer vision algorithms applied to the camera image to determine the location of the contact point. An advantage of FTIR based sensors over competing solutions (such as DI, DSI [115]) is that this technology suffers less from ambient IR noise, and is thus more robust to changing lighting conditions. On the other hand, it is well known that FTIR has some disadvantages:

- it does not sense finger proximity, the user must touch the surface;
- it is difficult to track the fingers during movements;
- though more robust to changes in ambient light, it still relies on a control over lighting conditions.

To partly address such issues we propose to take advantage of the shadows that the user hands project on the interaction surface. Our experiments show that such solution allows to effectively sense user interaction in an uncontrolled environment, and without the need of screening the sides of the multi-touch table (see Figure 6.1).

### 6.1.2 Tracking IR Shadows

Tracking infrared shadows to improve the quality of multi-touch interaction has been studied before. Echtler and co-workers [42] describe a system to sense

hovering on the surface, and thus provide pre-contact feedback in order to improve the precision of touch on the user's part. However the system they describe is based on a controlled IR lighting source above the table. In this sense their system exploits an additional artificial lighting source, increasing the dependence on the lighting conditions.

Our solution, as further described below, exploits natural uncontrolled light to improve the tracking algorithm. We take advantage of the natural IR noise to aid tracking, thus turning one of the main issues of MT sensors into a useful quality, making it possible to enhance tracking precision and implement pre-contact feedback. The proposed technology exploits the shadows projected on the surface by the hands of the users to improve the quality of the tracking system.

As said above, ambient light has a negative impact on the IR based sensors when the light coming from the IR LEDs is not bright enough to prevail on the background noise. However, the hands of the user project a shadow on the surface (that will appear as a dark area in the noisy background). Such dark area is easily tracked because it is almost completely free of noise.

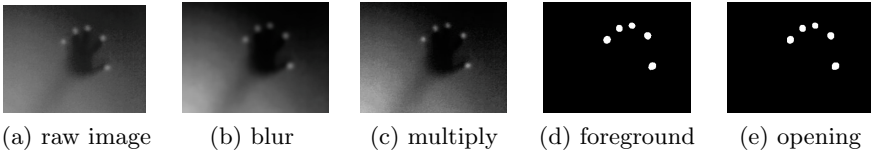


Figure 6.3: Smoothing, enhancement and foreground segmentation on IR light blobs.

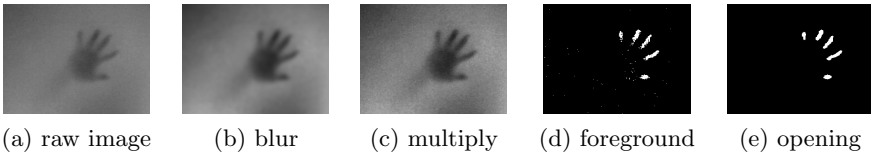


Figure 6.4: Smoothing, enhancement and foreground segmentation on IR shadows.

Furthermore, fingertips correspond to the darker parts of the shadow, and can be recognized with good accuracy. Note that tracking the shadow is more and more effective as the ambient light increases (as opposite from IR blobs tracking), thus IR tracking and shadow tracking tend to complement each other, the former working better in full darkness, the latter in full daylight. A second useful feature, consists in the ability of the shadow tracking system to sense objects that are only close (i.e., don't actually touch) the surface, thus allowing the sensor to recognize a richer collection of gestures.

Finally, a well known problem of FTIR based systems is that blob brightness decreases as the user moves her hands fast. This problem is typically addressed covering the screen with compliant surface and silicon rubber. Shadow tracking does not suffer from this issue, and can thus be exploited to improve finger tracking during sharp movements. Such complementarity is a key aspects of our work: it allows the system to work in less controlled environments, and to be more robust to changing lighting condition, as may easily happen in real world, off-lab installations. This latter is, as known, one of the major issues for computer vision based interactive systems.

Our implementation, based on OpenCV [121] for computer vision algorithms, shows significant improvements in the effectiveness of the sensor and, as a consequence, on the quality of interaction.

Figure 6.2 shows some frames from the image processing pipeline. Frames (1a-4a) are raw images as captured from the IR camera. The hand of the user is moving from top left to bottom right. Frames (1b-4b) are the output of the IR light tracking. Frames (1c-4c) are the output of IR shadow tracking. At (1a) the user has just touched the surface in an area relatively free of noise. The fingertips adhere well to the surface and the FTIR effect works perfectly as the result of IR tracking displayed in (1b) shows.

At (2a) the user is beginning to move her hand. As known, the IR light blobs tend to dim, but are still clear and trackable (2b). This is due to the fact that (i) the finger adhere less effectively to the surface while moving, and (ii) the hand is entering a noisy area. However the latter is partially counterbalanced by the IR shadow tracking (2c).

At (3a) the hand of the user is moving very fast and is within an area of high IR noise. The IR light blobs are invisible (3b), but the IR shadow appears clear and is easily tracked (3c).

Finally, at (4a) the user has completed the interaction phase and holds her hand still. Again the IR light blobs prevail on the noisy background and can be tracked with great precision (4b).

At this point, combining the two input sources (light blobs and infrared shadows) is a straightforward task; details are given in the next section (Tracking).

### 6.1.3 Image Processing Pipeline

As known, the process of finger tracking for CV based multi-touch sensors is typically modeled as a pipeline consisting of several stages: from image acquisition to preprocessing, finger detection and tracking. All transformations are implemented by means of convolution matrices. The steps through which our implementation passes are as following.

**Smoothing** A blur filter is applied to smooth the image removing the Gaussian noise, thus getting rid of pixel size spots (see Equation 6.1 and Figures 6.3b and 6.4b).

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6.1)$$

**Enhancement** A rectification filter enhances the luminosity of each pixel (see Equation 6.2 and Figures 6.3c and 6.4c).

$$img(x, y) = \frac{(img(x, y))^2}{(max(img(x, y)))^2} \quad (6.2)$$

**Background Removal Filter** The picture is filtered in order to find the areas of the screen on which an interaction is happening. To this purpose a  $7 \times 7$  matrix with Gaussian distribution was empirically determined. The result is matched against a threshold in order to select relevant areas. This operation in practice finds local maxima in the captured image. However the resulting image still presents some noise and must be further processed. Note that this same filter, applied to the negative image, is used in shadow tracking (see Figures 6.3d and 6.4d).

**Opening** An opening filter erodes spots whose size is smaller than a given value, often referred to as *salt and pepper* noise (see Equation 6.3 and Figures 6.3e and 6.4e).

$$img \circ m = (img \ominus m) \oplus m \quad (6.3)$$

**Lens Distortion Removal** The image is processed in order to compensate radial and tangential distortion due to the lens of the camera. Radial (Equation 6.4) and tangential (Equation 6.5) distortion correction require parameters  $p$  and  $k$  that can be computed by identifying distortions of images containing known regular patterns [16] (see Figure 6.5). Note that OpenCV provides black-box functions to this purpose.

### Perspective Distortion Correction

$$\begin{aligned} x_{\text{corrected}} &= x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{\text{corrected}} &= y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{aligned} \quad (6.4)$$

$$\begin{aligned} x_{\text{corrected}} &= x + [2p_1y + p_2(r^2 + 2x^2)] \\ y_{\text{corrected}} &= y + [p_1(r^2 + 2y^2) + 2p_2x] \end{aligned} \quad (6.5)$$

This last stage aims at transforming between capture coordinates and display coordinates and getting rid of perspective when (as often happens) the camera is not placed perfectly perpendicular against the plane of interaction. This



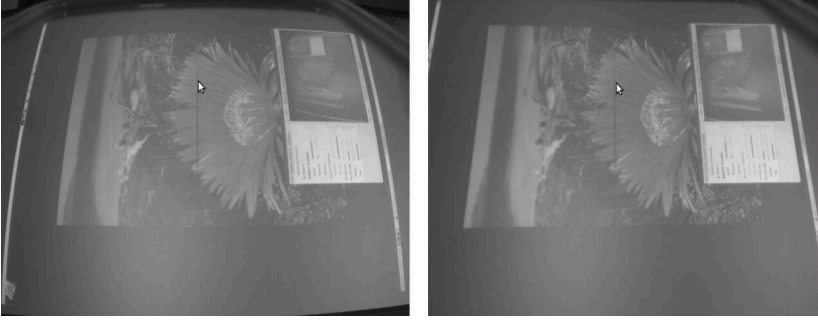


Figure 6.5: Correction of lens distortion (pincushion and barrel).

operation requires four points on the screen to be matched against 4 points in the capture. Usually this is performed manually (during an initial *calibration* phase). Such transformation is efficiently computed as an inverse mapping between triangular meshes [13].

To do so, the position of a point to be mapped from camera space to display space can be expressed in barycentric coordinates: if  $A$ ,  $B$  and  $C$  are the vertices of a triangle, a point  $P$  inside the triangle is uniquely identified by  $P = \lambda_1 A + \lambda_2 B + \lambda_3 C$ , where  $\lambda_1 + \lambda_2 + \lambda_3 = 1$ . Any deformation applied to the triangle does not change the baricentric coordinates of the point  $P$ , then since the coordinates of points  $A$ ,  $B$  and  $C$  in the display are known from the calibration phase it's easy to compute the coordinates of point  $P$  on the display.

The complete pipeline, both for IR blob light tracking and IR shadow tracking is depicted in Figures 6.3 and 6.4. See from left to right how the image is filtered to enhance meaningful features.

**Tracking** Finally, tracking fingers that touch the screen is done as follows:

1. an improved Continuously Adaptive Mean-shift algorithm (camshift) [121] is applied to determine a region of interest (ROI) surrounding the finger in each successive frame, in order to track the finger and reduce the region of calculation, the camshift algorithm constantly adjusts the size of the search window;
2. for each video frame, a matrix that represents the probability distribution of the foreground image is analyzed to determine the centre of the ROI;
3. the current size and location of the tracked object are reported and used to set the size location of the search window in the next video image.

4. based on the previous items, the system searches for fingers both in the shadow and light foreground images so that the tracking will continue even in variable lighting conditions.

All these allowed us to deploy a robust and reliable multi-touch table, easy to move and to calibrate.

## 6.2 Converting showcases and media facades into interactive walls

Space has always been a sensitive aspect in the designing of man friendly interactive environments. In the real world people usually demand a large enough space to live and work in, feeling trapped in a restricted space. This situation becomes even more evident during team working sessions, like meetings or workshops. If we observe shared open spaces, they need large spaces and long walls. The latter are for the majority, occupied by the instruments used in our day-to-day tasks like calendars, charts, maps and timelines. Most of these instruments are usually in a bit of a mess and people keep moving them around as they use them.

This is the reason why interactive walls are very much appreciated and suitable for hosting various fascinating and engaging applications that can be used individually or in team. Not only they act like a shared real wall but also display other types of multimedia contents, informing and at the same time, entertaining the users.

Our search for the most suitable technology to be used for the development of a shared screen was influenced by another important observation. When leaving an airplane, when we walk downtown, when we visit a museum we are constantly bombarded with information shown on a series of media facades displaying captivating forms and contents. Huge animated displays show dynamic images, commercials, they keep us up to date with weather news or the current time. Entire walls covered with encased monitors show video clips, trailers and various contents. These systems don't allow any kind of interaction, we are limited to indicate, watch or ignore them. Apart from the effective value of contents shown, we noticed that these big displays are positioned in places that are suitable for interactive use. Due to their width images are accessible to a number of passers by therefore are potentially perfect to be used to perform collaborative tasks. Even shop-windows, although not equipped with monitors, are installed in places where people gather to view their contents. In a normal day we get close to glass partitions, leaning over or touching the surface to get support or simply to show something. Once again communication is one directional, we can only watch toward the exhibition space.

In conclusion either the shops or the media facades were designed to attract a consistent number of people. It is a pity that these instruments, although

placed in suitable spaces, can't intercept us and do not capture our movements and indications. With these considerations in mind we put our efforts on the construction of an interactive wall that:

1. allows a multi-touch interaction and supports single user or multi user actions;
2. is easily fitted on any existing surface, like shop windows, displays, electronic billboards;
3. is modular and extensible so that can cover large surfaces;

Our goal is finding a technology that can adapt to different situations that can easily turn into interactive pre-existing environments and already installed surfaces, and that can be used to develop a number of interactive applications, from example showing stunning presentations or engaging visitors to a reception area.

### 6.2.1 Choosing the appropriate technology

Industrial capacitive and resistive technologies are used to produce multi-touch displays. Despite their accuracy and their widespread presence in a range of devices, such as tablets and mobile phones, these technologies are anything but cheap and make the construction of large interactive surface too expensive for our budget.

As already mentioned, in the last few years several researchers have worked on the construction of lab-made multi-touch displays. Most of them exploit Frustrated Total Internal Reflection (FTIR), Diffuse Illumination (DI) or Diffuse Surface Illumination (DSI) optical technologies [55] [115]. Those systems adopt high resolution IR cameras and must be placed in controlled lighting environments. Since they and other similar setups [33] [132] place the camera behind the sensing surface, the final touch-wall requires semi-transparent screens, prearranged structures and a certain amount of space. Thus, those systems are essentially used for the construction of closed-boxes or tables [116].

Other approaches [34,96] exploit cameras or sensors arranged around the screen while the position of the fingers is determined through triangulation. However these techniques need a meticulous arrangement of the sensors, require synchronized cameras and appropriate triangulation algorithms. Moreover the recent *zerotouch* [96] cannot manage large displays because of synchronization problems due to propagation delays in electrical signals.

Finally our survey has led to an innovative approach, named *t-Frame* [120], developed by the NIT's CRS4 team<sup>1</sup>. A t-Frame installation consists of a set of cameras placed on the top or bottom edge of the screen, facing down or upwards, as shown in figure 6.6. After a calibration step, the position of fingers is easily calculated by a triangulation algorithm, more simpler than others since all the cameras lie on the same screen edge (figure 6.6 b)<sup>2</sup>.

### 6.2.2 t-Frame technology

We now briefly summarize the t-Frame functioning. The first step consists in positioning a camera below the surface upward or downward. Then the user has to point manually an horizon in the background image. By analyzing the pixels that lie on this horizon we can detect when a finger appears in the image, breaking the line. Then we can draw a straight line that goes from the position of the camera to the point where this horizon is interrupted. By performing the same task with more cameras positioned next to the first, it is possible to calculate (or triangulate), with good accuracy, the position of the finger (image 6.6 b).

The cameras are not bounded to a fixed position or orientation, and can be arranged anyhow on the plane of the display. In general, with regard to the sensing performances, this approach disambiguates N finger-touches with N+1 cameras. Moreover it does not need any specific sensor or IR camera and even common webcams can be used. t-Frame requires less space than other technologies and can also cover large displays even when using multiple projectors. As we can read in researchers' reports, t-Frame was successfully adopted for the building of a 60" interactive display (figure 6.8(a)).

Despite the several advantages it boasts, t-Frame has still some limitations:

- the calibration step is manual since the user has to type manually some values and to point an horizon in the background image of each camera;
- the overall sensing resolution is fairly low and it cannot sense some finger interactions and fast movements;
- when using not synchronized cameras, the tracked finger/hand trajectory is very jagged;
- it does not discern between hand and finger interactions;
- it is not reactive enough if cheap webcams with low fps are used.

---

<sup>1</sup>Natural Interaction Technologies at the Center for advanced studies, Research and development in Sardinia

<sup>2</sup>These images are taken from the original publication [120]

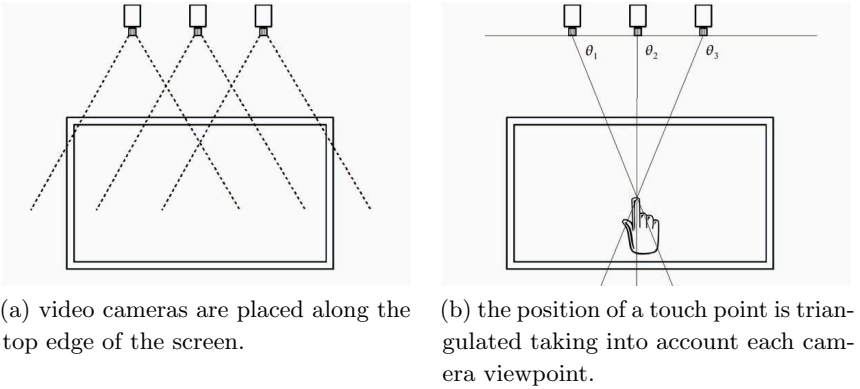


Figure 6.6: The overall setup of the *t-Frame* system, and a schema of the triangulation algorithm

With the aim of overcoming these problems, our work can be considered as an continuum of t-Frame technology. In particular, we report our findings and our improvements on cameras calibration and tracking process.

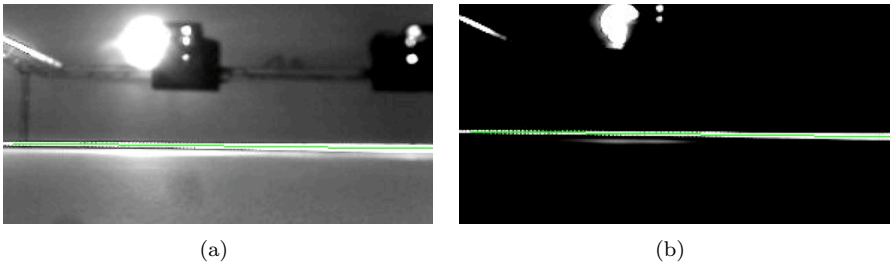


Figure 6.7: These figures show the *intra-frame* calibration. Some filters are applied to the original image (b) and the horizon is automatically calculated using the standard Hough transform.

### 6.2.3 Cameras Calibration

As introduced before, the calibration step estimates the exact position of each camera. We divided the calibration process in two crucial steps, that we called *intra-frame* calibration, and *inter-frame* calibration. For each part we report the old method and our improvements.

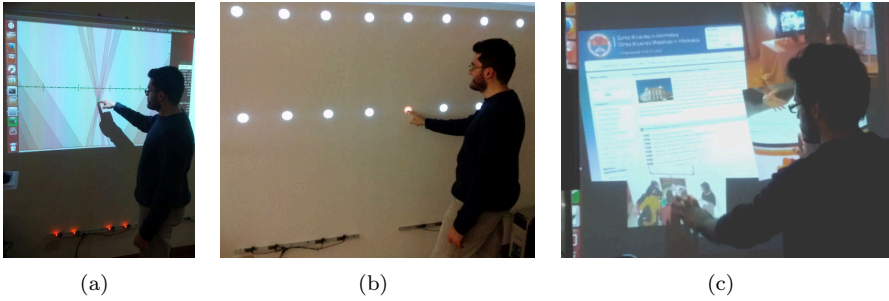


Figure 6.8: a) Finger triangulation when using multiple cameras. b) The cameras' position is calculated in the *inter-frame* calibration step. c) An interactive wall application.

### ***Intra-frame* calibration**

In the standard approach, each camera takes a snapshot of its field of view saving it as a known background and no touch is allowed; the user has to point an horizon in the background image of each camera. To avoid false contacts, the horizon must be specified as close as possible to the surface of the screen. Since monitors and projected screens have a regular shape, usually rectangular, our improvement consists in the automatic detection of the screen edge opposite to the camera (figure 6.7(a)), by using simple smoothing filters and the Hough-Lines algorithm [38]. The figure (figure 6.7(b)) shows the line corresponding to the detected display's edge.

### ***Inter-frame* calibration**

The aim of the inter-frame calibration is the calculation of all cameras' positions, analyzing the frames belonging to the different cameras and moving from the camera's reference system to the display's one.

In the standard approach, for each camera, the user has to touch with his/her finger three given points on the screen; this can be actually repetitive and tedious.

With our improvements, the separated calibrations needed for the different cameras are grouped in a single step. Therefore the user has now to press on a few points arranged in a grid that cover the entire screen. The modified system calculates automatically the number of points and their position according to the display size, better still to the number of cameras used. Since a camera partially covers the field of view of another one, only two points are needed for each camera (figure 6.8(b)).

## 6.2.4 Finger and Hand Tracking

Once the system is calibrated, the finger's position is estimated by triangulation. The next part should be fairly straightforward, we can track fingers by temporarily correlating their position among consecutive frames. However, we have to cope with two problems. The first is that when fingers are spread on the same place, the algorithm generates too many intersections (figure 6.9(a)), hence the final interface is unusable (figure 6.9(b)).

Secondly, when using unsynchronized cameras, the finger's calculated position is far from the real position because the cameras shoot the frame at different times. As we can see in the figure 6.11(a), the finger is moving from left to right. The camera on the right takes the image just before the left camera and this mis-synchronization generates a false finger position. This effect generates a very jagged finger/hand trajectory (figure 6.11(b)).

### Hand and Finger Disambiguation

In order to resolve the first problem we apply a density-based clustering algorithm (figure 6.9(c)), grouping fingers lying on the same area. This also helps detect false point of contact due to the triangulation process, because objects in these sparse areas are considered to be noise and border points. Not only the algorithm automatically groups fingers, but it can also distinguish between fingers and hands.

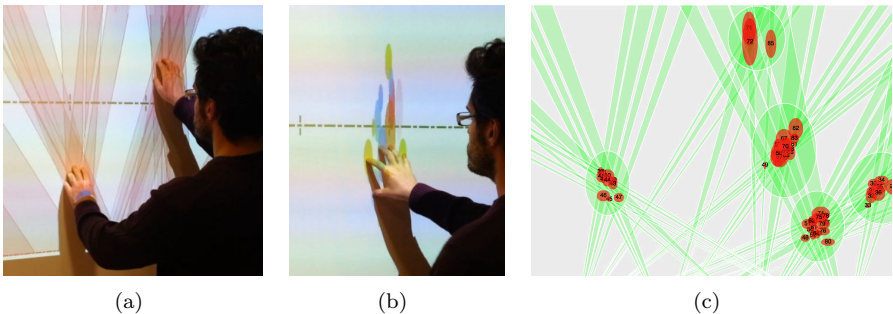


Figure 6.9: a) When fingers touch the surface on the same area, many intersections are generated. b) This effect produces a lot of false contact points. c) A clustering algorithm group fingers, associating each one of them to the correct hand.

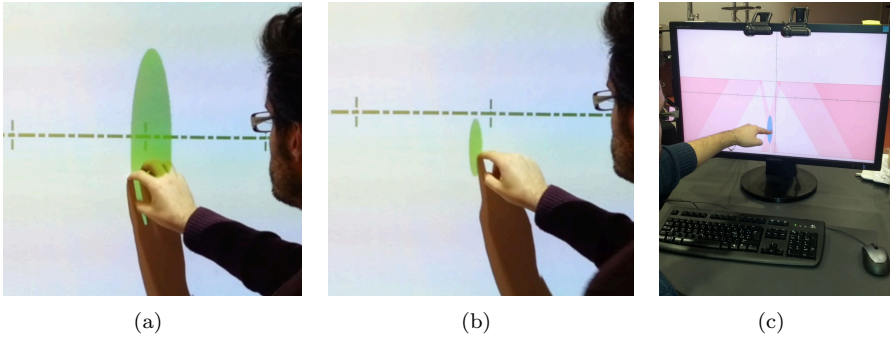


Figure 6.10: a) When fingers open, the hand is correctly recognized. b) When the finger are closed che algorithm triggers the event *hand-to-finger*. c) This approach makes a common desktop display a touch-screen.

### Trajectory Smoothing

In the tracking step we discard all the triangulated positions, only considering the clusters' position among consecutive frames. As described before, the tracking algorithm gives very zigzagged finger/hand trajectories. In order to polish these noisy fitting problems, we use a cubic spline with continuous second derivative, that is 1-dimensional curve fitting algorithm. In other words we apply a penalized regression spline for each x and y dimension. Then, the two 1-dimensional curves are joined together, creating a very smooth curve (figure 6.11(b)) .

### 6.2.5 Improved multitouch interaction paradigm

These improvements let us add other events to the traditional multi-touch interaction paradigm:

- *finger-to-hand* when fingers of the same hand spread open (figure 6.10(a));
- *hand-to-finger* that's when from spread fingers position you pull the fingers close (figure 6.10(b));
- *hand-up*, *hand-down*, *hand-move* when the open hand moves over the sensing surface.

### 6.2.6 Discussion

The final system tracks very smooth curves and the use of simple algorithms allows the development of fast interfaces (figure 6.8(c)). The installation not only



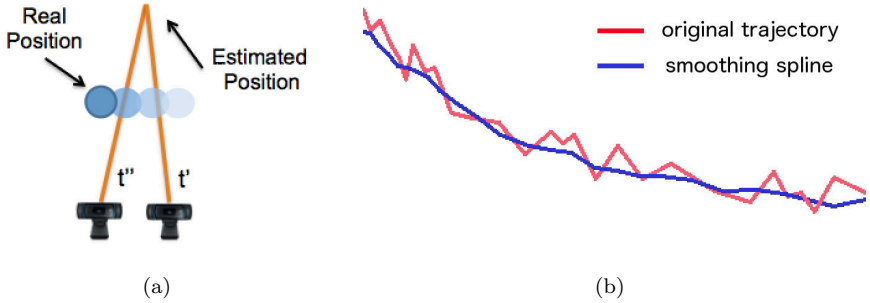


Figure 6.11: a) Using unsynchronized cameras, the estimated finger position is far from the real position. b) A regression spline is used to smooth down the zigzagged trajectory.

detects finger's positions, but also distinguishes between hands and fingers. This feature can be very important in the construction of interactive environments designed for co-operative and collaborative work.

Moreover, our improvements make t-Frame installations easier to reproduce and allow inexperienced people to use common webcams and monitors (figure 6.10(c)). In fact hands and fingers positions are estimated with a good accuracy even using unsynchronized, low resolution and low fps cameras.

### 6.3 Summarising

Having the need to plan interactive spaces that could host collaborative activities, we focused our research on the construction of two types of touch surfaces, a touch-table and an interactive wall. Through the improvement of optical technology the multi-touch table supports applications and teamwork even in unfavorable light conditions, typical of open spaces and busy places. We also examined in detail and fine-tuned an innovative technique for the development of an interactive wall. The improvements we introduced allow for the transformation of an existing visual surface like a window or a display into an interactive one; furthermore being the installation simple and modular, allows for the covering of entire walls and media facades.

Both techniques above described makes possible the designing and construction of interactive collaborative environments using easily available materials, therefore keeping the development costs low.

Up to the moment we wrote this dissertation, we did not have any technical assessment on the sensors performance. However all prototypes have been tested several times not only by different people during the development process, but have also left the lab were they were devised and constructed. They were in fact taken and installed in exhibitions and conferences seen by hundreds of visitors.

The complete list of fairs and exhibitions is reported in section 13.2.

Various interactive applications entertained the public that seemed to be enthusiastic to use our two solutions, the multi-touch table and the interactive wall. The first positive feedback encouraged us to keep with the good work.

In particular these multi-touch devices can be considered the basis of our work and, as described in the next part of this dissertation, we will consider them as a good environment where we can design and test manipulative and gestural interfaces.

**Acknowledgement:** This chapter is based on revised contents from the paper: Samuel A. Iacolina, Alessandro Soro, and Riccardo Scateni. *Improving FTIR based multi-touch sensors with IR shadow tracking*. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). ACM, New York, NY, USA, 241-246.

## Chapter 7

# Free-Hand Interaction

As already discussed, by means of manipulative and gestural actions people can interact and communicate more actively with a computer while a properly structured interface facilitates the control of a system. Free-hand interfaces are useful in this context since they can be used to exploit user movements detecting gestures and manipulations analogous to the way people interact with the real world.

This chapter describes the improvement of the free-hand interaction starting from the search for a congenial interface, suitable for the exploration of 3D contents. We focus particularly on the selection task when exploring 3D models through the use of a technique that allows the detection of a grasping gesture. Finally, we show the design and compare the tests on two alternative natural interfaces: multitouch and free-hand gestures. Both provide a natural dual-handed interaction and at the same time free the user from the need of adopting a separate device.

### 7.1 Natural exploration of 3D models

The design of the user interface is crucial to the development of hardware and software for the exploration of 3D models. Terms such as 'easy to use', and 'designed with your needs in mind', are often used to describe such technologies. With the recent explosion of off-the-desktop paradigms, such as virtual and augmented reality, ubiquitous computing, etc, the design of a 3D user interface becomes even more critical for researchers and developers. The user can feel manipulation and navigation of 3D virtual objects as difficult tasks, especially when using common interfaces. With traditional input devices such as mouse, trackballs, etc. the interaction doesn't insist directly on the models, but is mediated and requires a training period. However, many real world applications

don't allow any training before using of certain interfaces.

More and more cities, for example, are investing in the tourism industry, placing interactive information points in streets, squares and communal areas. We shouldn't get surprised these information points show 3D contents. The aim of these installations is to give useful information to visitors, offering at the same time a system that allows virtual exploration of architectural buildings or views of the city. This experience allows a better user involvement. In fact, people not only use menus and links to browse digital contents, that resemble the browser of a mobile phone or a computer desktop, but also interpret and interact with virtual environments.

This is why government institutions are actively looking for innovative and creative installations, where the relationship between users and content is more natural, thus forgetting there is a computer controlling their experience, or making them move away from preconceived ideas about it.

The aim of this chapter is the designing of interfaces that allow even inexperienced users to explore 3D objects through hand manipulations analogous to the way people interact with the real world. We report of two different interactive systems for natural exploration of 3D models by the use of multitouch and free-hand interfaces. Thanks to a natural 3D interface casual users can act on 3D objects with simple gestures and manipulation.

## 7.2 3D Interaction

In 3D interactive environments users can move and act in a three-dimensional space, both the user and the system work on information based on the position of objects in 3D space.

The place in which the interaction is performed can be either the physical space, a computer simulated representation, or a combination of the above. When user input is performed in the real space people can control the system by means of gestures or movements, captured by a suitable sensor.

In other words, the 3D interactive model defines a 3D space where users perform their tasks and share information with each other and with the system. Such scheme is intuitive since humans always interact in three dimensions in the real world [12].

Tasks can be classified in

- selection and manipulation of objects in the virtual space
- navigation

- control of the system

Such activities can be performed in a virtual space by means of different interaction techniques and using interactive tools. 3D interaction techniques are schematized according to the above classes of actions: techniques that support the exploration of the virtual world are defined navigation techniques; those ones supporting the selection and interaction with virtual objects are labeled as selection and manipulation techniques. Last, system control techniques support the activities of control of the application itself. In order for the system to be usable and effective, interaction techniques and devices must be tightly and coherently interrelated [10]. The power of interaction with a virtual model within the real world allows the users to exploit their natural and innate ability of manipulation, and to set in real world the exchange of information with real objects. Users, though, still face difficulties in the interpretation of the virtual 3D scene and in understanding the interaction paradigm [24]. Even if moving in a three-dimensional world is natural, difficulties arise since, unlike in the real world, the virtual environment doesn't allow the user to exploit all sensorial abilities: the ability to sense perspective and occlusions are primary senses used by humans. Furthermore, though the virtual 3D scene appears three-dimensional, it is still a projection on a 2D surface, which causes inconsistencies in the perceived depth, and misunderstandings in how the interaction should happen [78].

### 7.2.1 3D Interfaces

The user interface is the medium for user and system communication, they provide a device for the representation of the three-dimensional state of the system, and devices capable of acquiring the 3D input from the user (manipulation). The simple use of a 3D output device is not enough to provide a 3D interaction. Users must be allowed to perform 3D actions. To this end, specific input and output devices have been designed.

#### Hardware

3D input devices vary in terms of degrees of freedom and can be classified in standard devices, trackers, and gesture interfaces [12]. Examples of the first type include keyboards, stylus, joysticks, mice, touch screens and trackballs. Even a simple 2D mouse can be used as a navigation device if it allows the user to move within a virtual world. Trackers are capable of sensing and following the movements of the head, hands, body of the user and, given the position over time it is possible to update the viewpoint and the state of the virtual world. There are several types of 3D trackers, based on ultrasonic, mechanical, optical, hybrid inertial and magnetic technologies. Other devices, such as wired gloves and bodysuits can sense the position of the hands and body and send

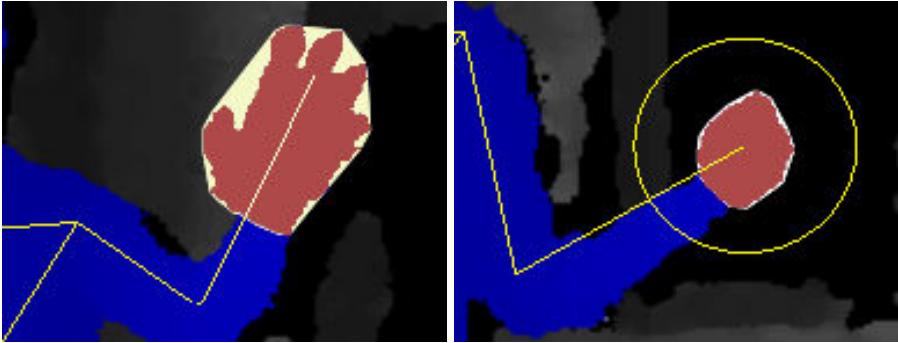


Figure 7.1: Automatic recognition of the shape of the closed and open hand: the red area is the segmentation of the hand, the white region is the Convex Hull.

such information to the system, enabling gestural interactions. Anyway these are still expensive solutions.

## Software

Users must be able to manipulate virtual objects. Manipulation typically consists of selecting, moving and rotating objects. Direct-hand manipulation is the most natural technique, because it is intuitive for people to use their own hands to act on physical objects. Most techniques involve a virtual hand to select and relocate virtual objects and adopt 3D widgets to modify the settings of objects or to search and move objects [20]. Other techniques exploit Non-linear Mapping for Direct Manipulation [107] and ray casting [88], in which a virtual beam is used to choose and select an object. Recent research focus on the design of interactive surfaces and whiteboards, to use e.g. in classrooms.

## 7.3 Our Proposal

### 7.3.1 Natural object exploration

A variety of devices could be used for motion control, the standard layout need a 2D vector, and a state (pressed/released). A simple approach without multitouch control is to use a two button 2D (or 3D) mouse, use mouse drag to specify motion, pressing one button for rotate/pan and another one for scaling. However, this standard scheme of object exploration, can be a difficult task for a novel user [11], even with a common 2D display . We introduce a 3D user interaction technique which allows casual users to inspect 3D objects at

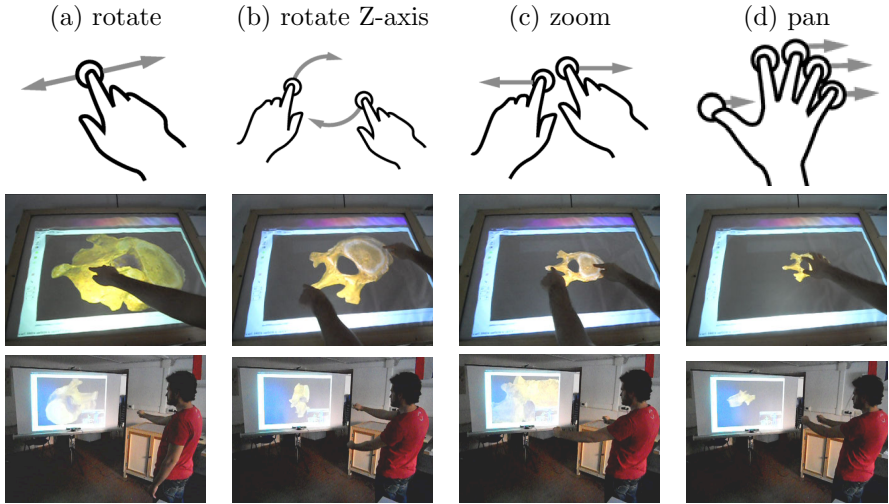


Figure 7.2: Comparison of the different manipulations performed on multi-touch table (row 2) and free hand manipulation (row 3).

various scales, integrating panning, rotating, and zooming controls into natural and intuitive operations. Our experiments concentrate on two innovative input devices: multitouch tables and vision-based gesture recognition, both allowing a fully unencumbered interaction. Much of user interaction is replaced by simple, natural hands motions, reducing user interface complexity and user burden.

### 3D Viewer

Our research includes the design of a 3D viewer. We have implemented several input filters for common file formats, both binary and ascii based, such as ply, off, x3d, vrml, etc. The interface is designed to support both precise input (such the ones the user performs with a traditional mouse) and more rough, but also more natural, actions such as those ones typically performed on multitouch tables. Tools for the computation of bounding box, centroid, high resolution texturing, normals etc. are also provided.

### Multitouch Experience

Multitouch systems try to fill the gap between physical and digital world, and provide a valuable support to the design of tools and environments in which the control is mainly physical. A multitouch system integrates the visualization (and consequently elaboration) of a model with its direct manipulation, providing

an interaction paradigm in which the place where the action happens, and where its effects are displayed are coincident. In other words the information is transformed into a physical object, the 3D model, that can be manipulated to change some of its properties, such as shape, size, position. We have readapted the traditional multitouch interaction paradigm to fit the exploration of 3D contents.

**Rotation** by touching the model and moving the finger to a given direction the user can control a rotation hinged on the barycenter of the object.

**Z-axis rotation** by using both hands it is possible to cause a revolution around the z-axis, where the X and Y axes are coincident with the vertical and horizontal sides of the screen, while the Z axis is perpendicular to the surface of the screen.

**Zoom** by using both hands and moving them to opposite directions the user can resize the model. Moving the hands apart from each other enlarges the model, while moving hands towards each other shrinks the model proportionally.

**Panning** hitting three or more points, even with the fingers of one hand, and dragging towards a given direction translates the model to the same direction.

Since the interaction is not limited to a single hand, the multitouch experience gives to the virtual objects some properties of the real world, i.e. the possibility of direct manipulation by means of actions that are natural and intuitive for the user, that is thus able to exploit his/her own abilities of gesturing and shaping the world with the hands.

## Free-hand Interaction

To further complete and improve the above paradigm for the exploration of 3D models we have extended it to support free-hand interaction. The model is displayed on a wide screen, under which a depth camera provides sensing of the user movements and gestures. Just like the multitouch interface described above, our free-hand interaction scheme is based on a press/release paradigm. Going back to the experiment on tangible interaction we carried on in Chapter 5, the possibility of using an action commonly used in the real world is of primary importance for the users. This is why we map the press/release scheme to the act of opening and closing the hand, which resembles the act of grasping a real object. The algorithm that recognizes the state of the hands is described further on.

**Rotation** closing the hand(s) and moving them along a direction the user can control the rotation of the model around its barycenter.



**Z-axis rotation** acting with both hands and dragging towards a given direction the user can rotate the model around an imaginary axis perpendicular to the surface of the screen.

**Zoom** operating with both hands, and imposing a movements along opposite directions, the user can change the size of the model. Moving the hands apart the model is enlarged, while moving the hands towards each other the model is reduced.

**Panning** dragging with both hands the user can move the object towards a given direction. Again, the ability to interact with the 3D models just like if they were physical objects, by means of both hands, lets the exploration of the models an immediate and intuitive operation.

Being able to manipulate 3D models just as if they were physical objects, by mimicking with both hands the real movements of grasping, rotating and moving in the real space, simplifies greatly such operations.

### 7.3.2 Description of the technology

A FTIR multitouch table [55] [115] has been used to support the evaluation of the multitouch manipulation of 3D models. The FTIR sensor was improved to allow further robustness to changing lighting conditions [67], explained in section 6.1.

In order to support a spontaneous manipulation, we developed a vision-based tracking algorithm based on the Microsoft Kinect depth sensor. We initialize the hands position detection with the skeleton tracking algorithm provided by the NITE framework [108], which also detects the hands in the depth image. Then, we incrementally track such positions in the subsequent images, recognizing hand open/closed shapes by estimating the local surface areas in the depth images (represented in red in Figure 7.1). The hand region is compared with its convex-hull area (represented in white): if the hand is closed, the two silhouettes are nearly coincident, and such ratio is close to 1; otherwise, when the hand is opened, the two silhouettes will consistently differ.

The communication between the 3D viewer and the multitouch sensor or the the depth sensor is supported by the TUIO [75] network protocol, according to the scheme in figure 7.3.

## 7.4 Summarising

People use their senses to gather and interpret the physical world, and the tools that exploit these abilities are the most effective. From this point of view, the most common user interface in HCI is based on different devices used for

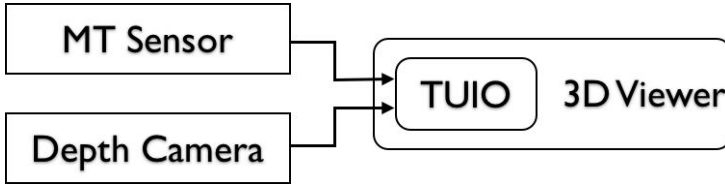


Figure 7.3: Schema of the protocol infrastructure.

sensing the command input and visualizing the output effect resulting from that command. Hence it stands between the physical world and its representation in a way that does not satisfy many practical cases. By contrast, a natural technique of exploration based on direct manipulation, positively builds on the coinciding input and output place.

We shown how multi-touch and free-hand manipulation allow inexperienced users to explore 3D objects. Panning, rotating, and zooming controls are by now simple operations even for newcomers.

Furthermore this experiment let us compare the direct manipulation offered by multi-touch interaction to a free-hand interface based on more manipulative actions. The gesture of grasping has proved useful for the task of selection and eases the exploration of 3D contents. We will use these considerations to design other interfaces and applications in the next chapters of this dissertation.

**Acknowledgement:** This chapter is based on revised contents from the paper: S. A. Iacolina, M. Corrias, O. Pontis, A. Soro F. Sorrentino, R. Scateni *A Multi-touch Notice Board Fostering Social Interaction*. Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI, p.13, September 2013, Trento, Italy. [65]

## Chapter 8

# A Unifying Framework

As seen in the previous chapters, multi-touch tables and walls permit a more natural exploration of contents through gestures and manipulations. Moreover free-hand interfaces are used to mimic with both hands the real grasping movements, manipulating the contents just as if they were physical objects in the real space.

Before developing and designing more advanced applications based on gestural interfaces, we have to build the instruments needed for the development of these interactive applications. In other words, there is the need of developing a software framework to build interactive environments supporting different gestural paradigms, such as multi-touch and free-hand ones. This framework can be used, for instance, to design simple interactive photo or video viewer, or stunning interactive notice boards.

### 8.1 Goals

The objective is the creation of a complete framework from different viewpoints. Regarding viewing capabilities, it is necessary to develop an efficient system capable of visualising standard contents like photos and videos. The different contents then, need to be loaded at runtime, accessing local and online resources. In case of a framework being deployed in multi-touch optical surfaces, it has to include a multi-touch software sensor, compatible with visual markers and other tangible supports. Finally, to support free-hand interaction, the system has to include a depth camera software sensor.

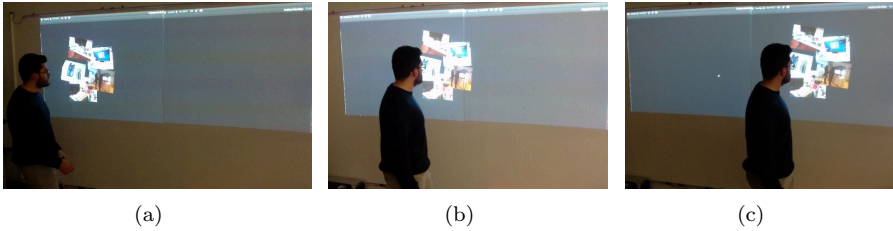


Figure 8.1: a) b) c) The contents are allocated according to user position.

## 8.2 Description

Our framework, originally designed for the testing of the approach described in section 6.1, evolved into the following characteristics:

**Standard contents:** it includes a viewer allowing the visualization of images, videos, maps, 3D models, text documents.

**On line resources:** in order to create demos with existing real contents we fitted the framework with a login system accessing the contents of the main social networks. Furthermore the system is capable of accessing 3D models online such as VRML and X3D.

**Remote control:** the framework sets in motion a network service that allows communication and upload of run-time contents. This is particularly useful for cellular phones, a device that is always with us as if it were a remote-control, using its inertial sensors to move the contents on the screen, the keyboard to enter text (figure 8.3(c)) and its wifi connection to communicate with the viewer.

**Fiducials and tangible objects:** in multi-touch environments, the framework supports visual markers, fiducials or other objects positioned on the surface. We describe the details in the following paragraph.

**Recognition of users:** in free-hand environments, when connecting a depth camera, the system recognises and identifies the various users to which the shown contents are associated.

**Automatic positioning of contents:** when a depth camera is connected, the system allocates the contents on the screen according to the user's position, as shown in figure 8.1.

To attract the users we provided the viewer with an appealing interface, creating visual effects and animations. As we will see in section 10.2.5, using a multi-touch table, thanks to this viewer a shaking animation and a translucence accompanies the *drag and drop* gesture (figure 10.4).

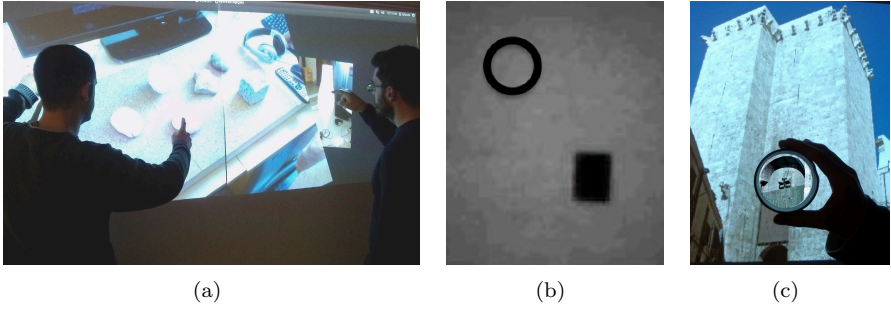


Figure 8.2: a) When using standard interfaces, a zoomed image covers another user space. b) Optical sensors can easily detect the objects laying on the table. c) Zooming an image using a *tangible* magnifying glass in a multi-touch table.

### 8.2.1 Interface compatibility

Our viewer supports multi-touch tables, interactive-walls, free hand interfaces and traditional desktop platforms.

### 8.2.2 Desktop computers

By using *adrag and drop* gesture or *copy and paste*, the user adds contents to the viewer. The system discriminates the contents based on the type, recognizing a video, an image or even a 3D model.

### 8.2.3 Multi-touch tables

Great effort has been devoted to the correct visualization of images and videos in a multi touch table used by different users. With the traditional multi-touch interfaces, a pinch to zoom gesture enlarges the entire image, as shown in figure 8.2(a). However, when zoomed, the image can cover the other user's working area. To sort this problem out we modified the multi-touch sensor described in section 6.1 so that could recognize hollow objects put on the surface (figure 8.2(b)). The viewer is sensitive to round shaped objects and enlarges the portion of image or video that falls within the object, as if a real magnifying glass was being used (figures 8.2(c)).

### 8.2.4 Free hand interfaces

Users can move contents displayed on the screen through grasping gestures. What's more, the system recognizes the different users, associating the contents

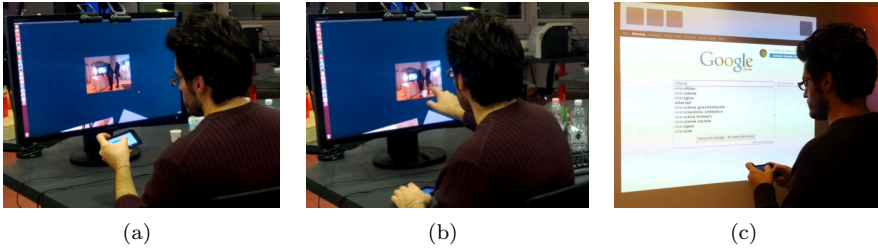


Figure 8.3: Sending a content from a mobile phone (a) to a touch-screen (b), the user can continue his/her personal work. c) The keyboard of a mobile phone is used to type characters in the interactive wall.

to each user, updating his position when he moves away from the screen (figure 8.1).

### 8.3 Implementation details

To get visual effects and enable a quick rendering at the same time, we developed some shaders and used a kd-tree to partition the graphical objects in the space. The Qt5 libraries and its declarative framework (Qt Quick) facilitated the creation of appealing animations, such as a video displayed in a box moving and rotating in the 3D space. This implementation allows to keep high the frame rate, above 100 fps, even showing many full HD videos and 3D models, although our computer is pretty dated, Intel Core 2 Duo and Geforce 280. To create a system where the various devices “talk” to each other we used the protocol xPlaces [35] to create a network infrastructure. In addition to this we changed this protocol implementing the division and reassembling of a packet that exceeds the UDP datagram dimensions.

### 8.4 Discussion and summarising

This chapter reports on a highly flexible framework that works with various interactive environments, extending their functionality. The framework can be used for the development of systems that need adapting to various configurations, such as multi-touch, free hand, desktop computer or hybrid systems. Thanks to its network capabilities, it is possible to create a system where the various devices communicate to each other. This means that we can build an interactive system distributed in the space, deploying multi-touch sensors, free-hand devices, and also the viewer in separated computers. Users can also use their own mobile phone using the keypad to add text, transfer material

from mobile to other interactive supports, i.e. to keep on working, as shown in figures 8.3(a) and 8.3(b).

Over the time this framework enabled us to carry out the experiments and demos, with the various interaction paradigms, described in this dissertation.







**Part IV**

**Gestural Interaction**



## Chapter 9

# Analyzing the Gestural Action

In this chapter we describe a few interfaces based on gestures resulting from technologies and paradigms described in the previous chapters. In order to examine the various aspects of gestural action, we here describe some of the interfaces devised for the resolution of specific and modest problems. Firstly, we focus our investigations on the browsing of visual documents, trying to design a gestural interface that overcomes the limits of standard interaction paradigms. At a later stage, we analyze the communicative aspect of gesticulation, designing a tool for quality evaluation of teaching in terms of gestures performed during an exposition.

### 9.1 Browsing visual documents by free-hand gestures

In the real world, the review of hardcopies is one of the scenarios where manipulative action prevails. On the contrary the browsing systems available in actual graphic interfaces offer an interaction based on limited movements with the mouse and the pressing of a key on the keyboard. Comparing for instance the number of hand movements we use to flick through the pages of a magazine to the browsing of digital contents like videos, documents or pictures using mouse and keyboard, we realize how personal computers are limiting our manipulation ability.

In this section we describe an interface that allows the browsing of digital contents through free-hand gestures with the aim of building an interaction that is closer to our manipulative abilities.

### 9.1.1 Description of the problem

The ease we create either textual or multimedia digital contents, is nowadays resulting in overproduction of contents. The circulation and sharing of materials, through Internet has pushed the state of things to the limit. The sharing of a folder or photo album with our friends is by now very common and we find ourselves searching, filing and retrieving an enormous amount of contents. It is exactly this huge amount of documents and their variety what causes the GUIs in our computers to come to critical crisis.

Computers have for a long time allowed the visualization of material of various types, like images, photo, videos, documents. Different interfaces exploit our visual memory and the human ability of recognizing images, like the visualization of a list of files or images or videos through previews or thumbnails. Furthermore, thanks to three dimensional animations introduced by cover flow interfaces [27], we can see files as if they were shown in a virtual showcase, and browse them visually flipping through snapshots of documents, website bookmarks, albums artwork, or photographs, by means of mouse gestures or keyboard inputs.

All these solutions facilitate the browsing process providing a better visualization of the contents, allowing a file search just like we would do in the real world, visually identifying them amongst the others. It has been widely demonstrated that visual search systems also support the content exploration of large document collections [32, 135].

However if we compare the interactions we perform on the digital contents when reviewing real documents, we realize that many of our manipulating abilities are not exploited.

In the real world, the review of hardcopy documents is one of the scenarios where the gestures and manipulative actions prevail. Let's think about the ways we search a photograph or a document. We usually spread the material on a table and shuffle with our hand to visually search until we find the photo or the document we were looking for.

In other words it is not only how these contents are visualized that counts, but also the way these interfaces are used. For this reason we created an interface that could not only present the contents with a more appealing format, but also designed for our inborn manipulative abilities.

### 9.1.2 Interaction scenario

As visual support we used a coverflow. Documents are set out in a line that occupies the screen along its width. The document we are viewing is at the center of the line, facing up. Other documents are slightly flipped towards the centre of the screen in an angle that allows a glimpse of its content but at the same time the highest number of documents are stacked close together.

As a device input we chose *LeapMotion* [98]. As described in section 4.6, it is a computer hardware sensor device that supports hand and finger motions as

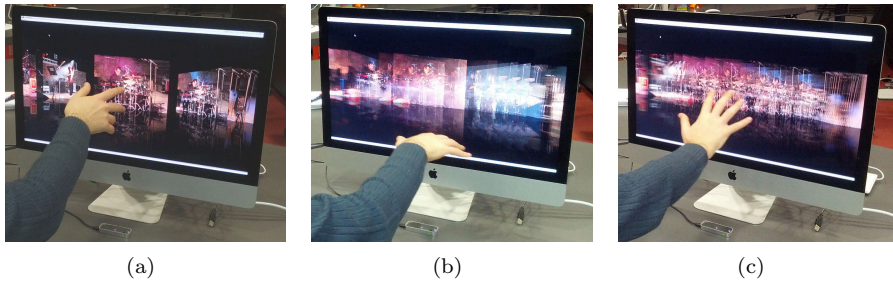


Figure 9.1: a) A swipe gesture allows access to next or previous document. b) Scrolling the list of thumbnails. c) A shuffle gesture randomly changes the order of visual documents.

input, comparable to a mouse, but requiring no hand contact or touching.

## Gestures

The proposed interaction consists of three simple free-hand gestures that allow the exploration of a series of contents at different scales.

**Swipe** As figure 9.1(a) shows, a simple left or right swiping gesture allows the browsing of an item list in one direction or the other, mimicking the action of throwing a sheet from the table and move to the next document.

**Scroll** Spreading the fingers of one hand in horizontal position in front of the display and moving the hand to the left or right, the items can flow fast following the hand position in the real space. It's like the hand would be an imaginary scroll bar up in the air (figure 9.1(b)).

**Shuffle** To mimic the shuffling of hardcopy documents like we do in the real world, we implemented a hand shaking gesture that changes at random the document's position (figure 9.1(c)).

### 9.1.3 Discussion

We can say that the poor manipulation offered by traditional interfaces limits our ability of visual search and image recognition. In the real world we use our hands to find a document we are looking for. To offer a more exciting exploration of visual documents we implemented some simple gestures. Not only people can easily mimic them but they are already familiar with using

them in daily tasks.

While the swipe gesture is the most discrete one in our vocabulary, the scroll gesture is a combined one, it implies an open hand gesture and a manipulation, with a continuous movement in the real space. With regard to shuffling, when maintained over time, documents keep changing position.

### 9.1.4 Applications

Browsing visual documents by means of free hand gestures can be useful in a range of different application fields, where the touchless is definitely needed. It could be used in hospitals for instance, to view x-rays and CAT scans, leaving hands free and documents clean.

Let's also think about interactive information points distributed around in a city, touchless screens would not require tourists to take off their gloves for instance, also avoiding dirt or germs' contamination.

## 9.2 Web based Video Annotation

In the first part of this dissertation (section 4.3) we highlighted the importance of gestures. Just because they help the exposition and they have a large communication load, gestures play a crucial role in teaching, because they encourage audience attention and learning. Actually, people are more inclined to learn when teaching is accompanied to gesture: explaining the same concepts in two or more different ways helps to better understand it, when compared to using only the speech. The teaching supported by gestures encourages especially children to repeat on their own the same gestures made by teachers. Particularly, math teachers do different gestures while explaining to children a problem resolution. For example, when teaching to children how to resolve a simple math equivalence ( $7 + 6 + 3 = \_ + 3$ ), teachers produce gestures that convey to children the strategies to resolve this type of problem. When is asked to children to resolve this type of problems, they are brought to use, on their own, the same gestures made by teachers during the teaching phase, showing to have exactly learned how the strategy works [30].

We could think of evaluating the quality of the presentation from the gestural communication aspect. Such evaluation will help in analyzing speakers' teaching and communication skills, in order to help them improving the overall quality, focusing on performance strength and flow. We describe in this section the design, development and initial evaluation of MORAVIA (*MOTion Recognition And Video Annotation*): a collaborative web application for (semi)automatic gesture annotation. Extracting the body skeleton, MORAVIA detects position, movements and gestures of a teacher using a depth camera, such as the Microsoft Kinect<sup>TM</sup>. Then, our web application for video annotation allows collaborative review and analysis of the different video sequences. This is useful to both

domain experts, as a research tool, and end users, for self-evaluation. Finally, the overall system will be able of giving a quality score to the entire performance.

With regard to gesture recognition, section 4.6.3 described the most relevant proposals in the different fields touched by this experiment.

### 9.2.1 Video annotation

Videos have a high communicative potential, and therefore they are used as tools for acquiring knowledge. It was the availability of cheap video-recording to foster new research on gestures and today we can expect a similar explosion thanks to the introduction of automatic gesture recognition on videos.

Several platforms exist to support researchers in the analysis and annotation of videos, among these we describe here those two that most influenced the design of MORAVIA. *VideoANT* [61] is a web based video annotation tool, characterized by a minimal user interface, it allows free text annotation, and is often adopted for collaborative annotation tasks. However it lacks several interesting functionalities like annotations downloading and a users system. *Anvil* [79] is a desktop annotation tool which offers multi-layered annotation based on a personalized coding scheme. It provides very useful features such as color highlighting for annotations and coding agreement analysis. However, being a desktop based application, co-annotation and project sharing is not always straightforward.

### 9.2.2 MORAVIA

In our context a working group, that may consist of students, teachers or researches, collaborates to the annotation of a video marking significant moments. The video typically contains a teaching session that has to be evaluated. Sessions are recorded using a video camera, and then subsequently they are analysed in order to identify weak points and to suggest improvement. Through this technique, teachers operate an observation on themselves from the professional point of view, becoming aware of the manner in which their competencies are manifested, and manage to identify possible elements that interfere or hinder the training method. Extending this protocol to group evaluation allows to gather many different points of view, so leading to a more effective evaluation.

A further improvement, and our original contribution, is then to exploit, in addition to the plain video, the information on subject's body movements and postures captured by a depth camera (and thus suitable for automatic elaboration). Our proposal consists, as already anticipated, in a tool for quality evaluation of exposition in terms of gestures: this involves the creation of a

classification model that, taken in input a video recording containing a speaker who performs an exposition, is able to detect different gestures performed by the same speaker and is able to give a score to the performance.

Now we describe the steps necessary to achieve this goal.

### 9.2.3 Training set

The classification model should be trained starting from a training set, that in this case, given the nature of the problem, is composed by several types of gestures and an expositive score associated with each of them. Because we did not have a training set of this type, it was necessary to build it by ourselves: to do gesture capturing we preferred to use techniques of video-recording combined with Computer Vision instead of techniques based on wearable sensors, because these last tend to be more intrusive in the exposition.

We decided to use *Kinect* that, as explained earlier, implement good quality sensors that facilitate the use of techniques of Computer Vision for the recognition of movements by identifying the human skeleton, providing a good enough performance. We expect that the affordable price and the good performance will make it the de-facto standard in a short time and will stimulate a renewed interest in gesture recognition research. Once we captured the gestures, we needed an evaluation about them; those evaluations, to be reliable, must come from experts on educational and psychological domain.

The best way to obtain evaluations is to collaborate with a group of experts. We contacted a group of experts in didactic valuations, that was already executing video recording of expositions.

### 9.2.4 MORAVIA: Video-Annotation Web Application supporting collaboration

After an analysis of existing video annotation tools and having discussed about it with the group, we decided to develop a video annotation software on our own. We identified the following features as essential:

- ease of use and minimal UI; since MORAVIA users may be very different in computer literacy;
- web interface for collaboration; the workgroup may be (and actually is in our case) spread in several departments/cities;
- possibility of downloading annotations to work offline;
- authentication of users; it is necessary to distinguish the attribution of any annotation;





Figure 9.2: The main video page of MORAVIA.

- customizable annotation structure; from the simplest plain text note to a very detailed annotation convention;
- support for common video formats, including *Kinect* ONI format;
- extensibility to include automation filters (such as HMM gesture recognition).

None of the platforms available today have all these features. In figure (Figure 9.2) there is a screenshot of MORAVIA with the various parts highlighted: part 1 contains the page header with site navigations commands; part 2 contains RGB video and *Kinect* vision of the current video: this part is a Mockup, at the moment only RGB video can be viewed; part 3 contains the annotations markers and bars: the multi-colored upper bar is still a Mockup, it will reveal with red color video sections where gestures are frequent; part 4 contains additional video controls; part 5 contains annotation management buttons; part 6 contains the textbox to insert new annotation; part 7 contains currently available annotations for the current video. At the moment, the site provides a simple authentication and users system, with a permissions subsystem for videos and annotations; furthermore it provides multi-language support and currently it supports English and Italian languages.

The cooperation with the group included our presence during their classic video-recording sessions: we placed the *Kinect*, paired with a notebook PC, along with the classic video-camera owned by the group. Thus the design of the system was refined and adapted to support the field-work as described so far. Also we managed to collect recorded expositions paired with the skeletal tracking of the speaker, to use as a training testbed.

Once we collected enough expositions samples (with related skeletons and video annotations), we proceeded with the definition of the evaluation model by using the techniques of gesture recognition and gesture classification already exposed in the state of art, and the association of scores to different gestures extrapolated from records provided us by the experts.

### 9.2.5 Drawbacks

Among the various difficulties encountered so far we can certainly mention the issues related to video-recording: in addition to classic problems of privacy and loss of naturalness in exposition due to the presence of a video camera, the group of experts often found difficulties to get teachers willing to be filmed and sometimes those which have given the availability gave up at the time of registration. Also, the registration with *Kinect* may cause further loss of naturalness since: it tends to be more cumbersome than standard video-cameras (has to be connected to a PC); it has to be placed closer to the speaker, and the speaker herself have to do a calibration pose of a few seconds necessary for the initial identification of the skeleton.

The group is also doing evaluations mainly on primary school teachers, with a series of additional problems. There are logistic problems, since classrooms sometimes are narrow, it is then difficult to obtain optimal positioning of video-recording; children are distracted by the presence of video-camera and *Kinect*. Issues of privacy are more delicate because of the presence of children, which are sometimes rowdy, and go on purpose in front of the video-camera to get filmed.

### 9.2.6 Discussion

In this section, we demonstrated how gestures are important in human communication and particularly during expositions, and we proposed the implementation of a tool for exposition evaluation from the gesturing point of view. We disclosed the various advancement stages needed to accomplish this objective, our actual progress of the work and the problems encountered. Our web site MORAVIA is continuously updated and already offers a basic functionality that manages annotations; we're working on the skeletons obtained from expositions recorded

with Kinect, that will soon permit the creation and development of a training set.

**Acknowledgement:** This section is a revised version of the paper: Marco Careddu, Laura Carrus, Alessandro Soro, Samuel A. Iacolina, and Riccardo Scateni. *MORAVIA: A video annotation system supporting gesture recognition*. SIGCHI - CHIItaly 2011 Adjunct Proceedings, 2011. [90]



# Chapter 10

## Cooperation

In traditional multi-touch interfaces, many interactions are based on discrete elements: we open a folder with a finger touch, like it was a push button. Discrete actions are the direct result of early computers and command line interfaces, when people used to push buttons and enter commands. Are we sure we still like this? In the real world, we constantly perform continuous actions that come naturally like opening a drawer or a door. Since nowadays computers are capable of reacting even to manipulations, couldn't we develop interfaces based on our abilities?

In this chapter we talk about an alternative interface that exploits direct and continuous manipulations, instead of discrete gestures, to explore containers, such as folders, groups and so on. Our application is designed for a bulletin board, where a user can pin a note or a drawing, and actually shares contents. The proposed manipulative interface is designed to support the presence of different simultaneous users, allowing for the creation of local multimedia contents, the connection to social networks. It provides a suitable working environment for co-operative and collaborative tasks in a multi-touch setup, such as touch-tables, interactive walls or multimedia boards.

### 10.1 Manipulative and gestural experience

Multi-touch systems aim at integrating the visualization with direct manipulation of symbolic objects that represent the information. As such, these systems are suitable to build natural interfaces based on the Objects, Containers, Gestures, and Manipulations (OCGM) metaphor [46]. Recalling what we have learned in Chapters 3 and 4, these interfaces are composed by objects representing metaphors for units of content or data, by containers that are

metaphors for the relationships between contents, while gestures are metaphors for discrete, indirect, intelligent interaction, and manipulations are metaphors for continuous, direct, environmental interaction. We have also illustrated the conclusions reached by George and Blake [46]: since the cognitive skills on which OCGM is based are developed very early [72, 122], interfaces using OCGM are more innate and natural, and they will have a lower cognitive loading and use skills-based behaviors.

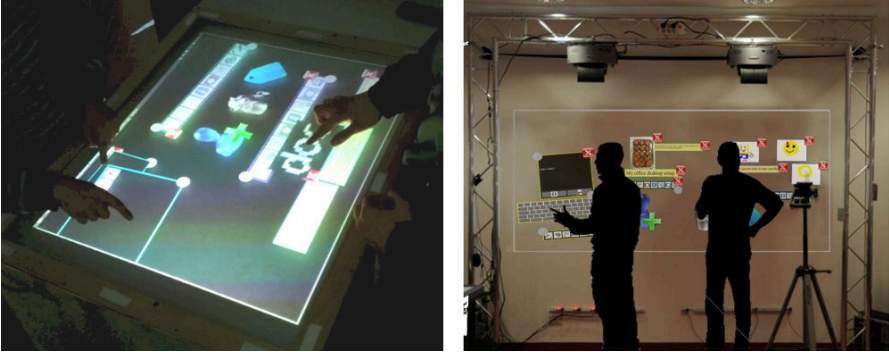


Figure 10.1: Touchtables and interactive walls are suitable working environment for cooperative and collaborative tasks.

In the attempt to find a flexible enough interactive system to support the implementation of context- and content-appropriate concrete multi-touch interface, also suitable for co-operative and social tasks, we have analysed the different forms of touch systems. The most popular obviously are mobile devices, smartphones and tablets, capable of connecting to social networks but not to supply simultaneous access to multiple users from the same device. In other words, these devices are personal and actually designed to be used by a single user. On the other hand, following seminal work from, among others, Buxton [22], and up to the recent developments [3], touch-tables and walls, whether they are *home-made* built or are commercial platforms, are used to supply multi-user interaction. Low-cost tables with a large display area, or even larger interactive walls combined with a OCGM interface have been adopted to create a suitable working environment for computer supported cooperative work, leveraging the exploration of new frontiers of social computing.

Despite multi-touch tables being inherently multiuser, the design of an interface that provides the support of truly multi-user applications is still problematic [100]: it could include the instruments for helping both multi-user and single user tasks without having accidental interferences between tasks of different users.

This work is focused on the development of an OCGM interface that, with the aim of supporting cooperative and collaborative work in a multi-touch

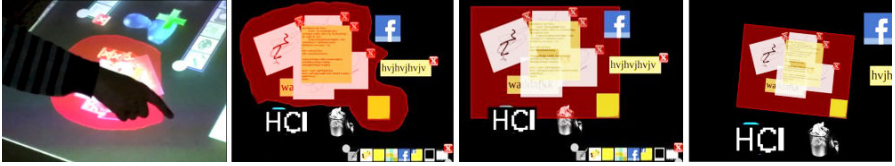


Figure 10.2: Defined by a generic shape, convex, concave and even intersecting polygons, a group is useful to simultaneously act on multiple objects.

environment, supports a multi-user interaction and allows for the creation of local contents, the connection to social networks, giving to multiple simultaneous users the possibility to create their own workspace containing social, multimedia, and interactive items. In our OCGM interface we introduce also a different paradigm for the interaction with the folder container. Based on direct manipulations, our solution supports the opening and the closing of a folder and the exploration of its contents by means of a continuous action of zoom in and zoom out. This perspective marks the distance between the proposed solution and other classical interfaces, including WIMP and standard OCGM interfaces, where opening and closing commands are activated by means of discrete gesture, usually processed after the interaction is finished; the output effect, generated by the command, can start only after the gesture is completed and properly recognized. Instead, in our work, the folder is an interactive object that can be directly manipulated, composed by the documents representing its content.

The rest of the chapter is organized as follows: we first describe in detail the communication with social network, then we detail the design of the proposed interface, with special attention to folder exploration; we, then, describe multi-user management and we report on some tests and interviews; to conclude, we discuss open issues and future work.

## 10.2 An improved OCGM GUI

Applications for multi-touch tables and walls often take inspiration from well known tools for communication and collaboration, such as bulletin boards, workbenches, blackboards. The bulletin board, on which people can pin notes and share ideas, messages, sketches or rants has been the model for our application. In order to design an OCGM based system, the proposed interface includes self-defined objects, relative, for instance, to textual content, images or videos, and containers, allowing also users to explore and manipulate selected aspects of their Facebook accounts.

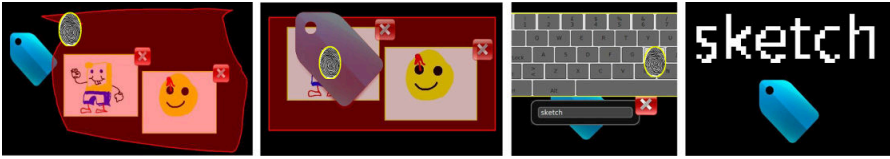


Figure 10.3: Once a group is dropped on the tag object, the user inserts a name and the folder appears like a simple tag or text.

## 10.2.1 Objects

We started designing objects that are responsive to touch events of finger up, finger down and finger move. Reacting to these events, all the items can be rotated, scaled, translated along the entire surface, giving the possibility to the user to move around the table perimeter and to continue his work. This approach brought also to the development of several interactive objects, including:

- base objects: postit, image and video;
- objects creating other objects: note editor, draw editor, toolbar;
- objects deleting other objects: trash.

In order to support the freezing and the subsequent restoration of the user task, some data, such as note and drawings, can be locally saved in the filesystem as text files and images. To further design a more complete system, the user is able to delete the unwanted objects using the trash object, removing them from the scene and simultaneously erase the related local data. In addition, the toolbar item is designed to allow the user to access all the functionalities defined by the interface. Creating all the mentioned objects is useful to open saved notes and drawings, display the editors and download social data. Providing an alternative, more interactive, creation and exploration of social contents, our application is able to push out the interactive objects, publish a text as a profile status or upload drawings as user images. Vice versa the user's status, images and videos, can be downloaded and converted into interactive objects.

## 10.2.2 Containers and Direct Manipulation

We also designed some items, such as folders and groups, as containers, that are the *grouping* of the objects expressing the relationship between the same objects.



### 10.2.3 Groups

Implementing groups is useful to be able to simultaneously act on more than one object, accelerating some tasks, such as the operation of deleting several objects or the creation of folders. To create a group, as depicted in Figure 10.2, the user must point the finger in the surface starting to track a line around the widgets should be grouped. The polyline can describe any shape: convex, concave or intersecting polygons are permitted.



Figure 10.4: A visual feedback helps the deleting task.

### 10.2.4 Folders

A group can be dragged and dropped above another item, the tag object, to create a folder container (Figure 10.3); when the drop action is finished, a text-box is displayed, giving the possibility to insert the folder name. Once created, the folder appears in the scene as a short text, one or some words representing the name inserted before. To support a more natural interaction, it is possible to open the folder with a simple and continuous zoom-in manipulation (Figure 10.5). When the zoom manipulation begins, the pixels of the text start to move and enlarge their sizes, and the user realizes that the pixels actually are the documents contained in the folder. Enlarging the object, the documents will be placed in a grid, therefore the folder content can be easily explored. Once the user has found the desired document, he can quickly open the item clicking on it. On the other hand, an operation of reducing the folder size by continuous zoom out cause a reverse movements and the shrinking of the documents, bringing the folder back to the original form of simple text.

The proposed interface differs from standard multi-touch interfaces, where the interaction with objects is predominantly based on gestures, processed after the interaction with the object, such as pinch to zoom or clicking. In these systems the folder is opened and closed by means of a gesture, a discrete action, because those actions express commands. We have tried to replace

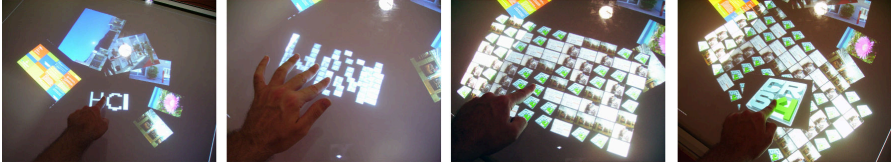


Figure 10.5: The folder container is explored by means of a continue manipulation of zoom in, while an inner content is opened pressing on its image.

the opening/closing commands with a different event, a direct manipulation, which is, by definition, continuous. Therefore in the system there is no state that expresses a folder opened or closed: the user can interact with contained documents once the zoom manipulation starts, and the contained items can be opened even if the folder is semi-closed.

### 10.2.5 Gesture and Visual Feedbacks

Addressing partly the issues that affect standard interfaces, in order to provide a visual feedback when an interaction starts, objects and containers reacts to manipulations and become semi-transparent on moving; therefore, a user is able to look under a specific widget and he can move it with precision along the entire surface, or even over another item.

We added to the standard gesture of clicking on an object, a particular behavior for the drag and drop gesture: in order to focus the attention of the user on the couple of interactive objects, pointing out the relationship established between the moved, dropped item and the underlying object, our interface activates a shaking animation as soon as the two objects collide. This visual feedback enhance the feeling that the drop action causes an effect that put in relation the two items. This feedback is useful, for example, to help the task of erasing items (Figure 10.4), allowing the user to delete an unwanted object that will not be used anymore by means of a simple drag and drop operation above the trash object.

## 10.3 Multi-user management

An appropriate multi-user interface should include all the instruments helping the user to manage the contents he or she intends to manipulate, establishing and preserving a relationship between the contents themselves and the user who created them. To reach this goal, trying to avoid a cluttered workplace crowded of items where the user is not aware of his work and where the system usability decreases during time, we introduce the concept of *workspace*. A workspace,

in this context, is the space where the user has worked, defining the set of all the items, objects or containers the user has created. In order to visually represent a workspace, the interactive items belonging to the same workspace are color-coded sharing all the same colored border. Providing a way to help the user to discriminate his/her own items, the interface allows to visually identify two workspaces belonging to different persons (Figure 10.6) and, in this way, the user can keep track of the items he or she has created and manipulated.

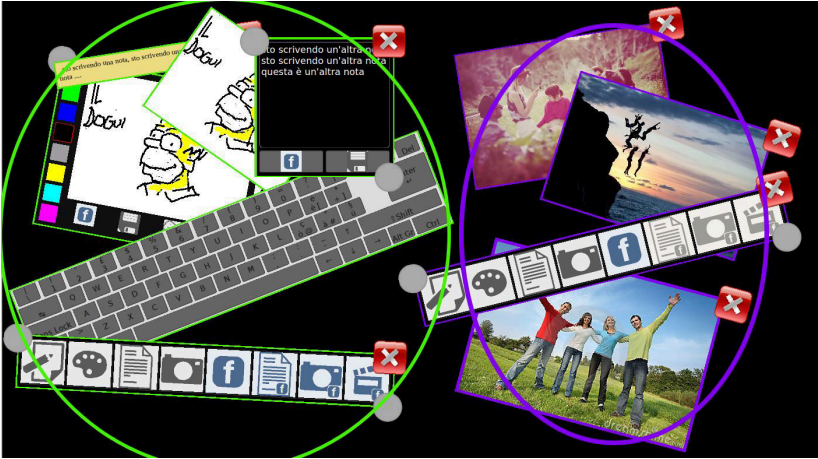


Figure 10.6: Our interface allows the users to visually discriminate two different workspaces.

### 10.3.1 User Interface

We designed and built our interface having in mind a set of goals:

**Provide a tool for logging in** The scene includes an object button in the background allowing a user to log in, the logging process displays a toolbar and, thus, the user can start working.

**Discriminate among different workspaces** It generates a random color whenever a new user logs in, the border color of each item visually helps the workspace identification.

**Manage multiple keyboards** In order to provide a scene where there is only one keyboard per user, a new keyboard is created when a writable object is opened for the first time and automatically closed when the last item needing it is closed; if multiple items need the keyboard, the interface manages the focus.

**Provide a way for logging out** Closing his/her toolbar a user can logout and destroy the sessions.

### 10.3.2 Technical details

To provide more details, the multi-user management includes:

- Establishment of multiple user sessions with social networks automatically reconnecting whenever the session expires.
- Management of local data, such as notes and drawings, providing a way to resume the work previously saved.
- Creation of an items hierarchy; starting with the opening of the toolbar at user login, the interface's items are kept in a hierarchy where the root item is the toolbar, so that it is possible to easily go back to their owners.

## 10.4 Preliminary tests

An improved FTIR multi-touch table, described in section 6.1, has been used to support the evaluation of the multi-touch manipulation and to test the interactive interface. Describing an usual task, a new user starts to work, by clicking on add-user button. After the login phase a toolbar will be opened, in which the user can insert his own credentials, a personal toolbar will be opened. Meanwhile other people are already working at the same time, on the same device, in their own working area. Hence the user can work independently, or in a collaborative way with the other users exchanging local or social contents like images, videos, and text. After early tests, we have performed some preliminary semi-structured interviews, verifying that the proposed interface is particularly suitable for large scaled multi-touch surfaces, such as home-made touch-tables and interactive walls, and provide a suitable working environment for computer supported cooperative tasks.

## 10.5 Discussion and summarising

What we learnt in the previous parts of this dissertation is finally used here, starting from the adoption of an reliable multi-touch sensor devices, the definition of a co-operative environment, up to the exploitation and comparison of manipulative and gestural actions.

Exploiting direct manipulations in order to supply a more natural exploration of containers content, we proposed an interface designed for a multi-user environment (Figure 10.1), allowing the creation of multimedia contents, the

exploration and manipulation of these contents as interactive objects and the sharing through social networks.

The present study is relevant to different application domains like the education sector involving multiple users and enabling co-operation and co-operative learning; in museums, for example, touch-tables could support visitors making the availability of information on artworks and scientific inventions more pleasant. Other important application fields could be the rehabilitation one, to support the patient, the entertainment industry, or the press and weather forecast.

**Acknowledgement:** This section is a revised version of the paper: S. A. Iacolina, M. Corrias, O. Pontis, A. Soro F.Sorrentino, R. Scateni *A Multi-touch Notice Board Fostering Social Interaction*. Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI, p.13, September 2013, Trento, Italy. [65]



# Chapter 11

## Evaluation

As afore mentioned, natural user interfaces are often described as familiar, evocative and predictable, based on common skills. Though unquestionable in principle, such definitions don't provide the designer with effective means for creating a natural interface or evaluate a design choice against another. Various important issues in particular are open:

- (a) how do we evaluate a natural interface, is there a way to measure 'naturalness'?
- (b) can natural user interfaces provide a concrete advantage in terms of efficiency, with respect to more traditional interface paradigms?
- (c) which kind of user interface prevails among others?

In this chapter we discuss and evaluate observations of user behaviour with the intent of comparing various interaction scenarios. Obviously the performances of two or more different interfaces strictly depends on the type of application. We can't affirm that interfaces based on free-hand interaction, for example, are generically better among others in whichever scope. For this reason, we take under review specific tasks or applications in order to evaluate and compare distinct interfaces.

In the first section we observe the task of pair programming, performed at a traditional desktop versus a multi-touch table. As demonstrated in the previous chapter, the main advantage of multi-touch tables and walls over desktops is their being inherently multi-user: people cooperate to the task at hand, sharing or negotiating the use of the device in a natural manner. Furthermore in multi-touch environments people feel encouraged to exploit their own manipulative and gestural ability. The importance of

non-verbal communication is highlighted due to the fact that it has a positive impact on many cognitive processes. With this experiment we intend to verify whether when using a multi-touch setting there is an increase of non-verbal communication (gestures, body postures, facial expressions, etc.) than when adopting a traditional desktop.

In the second section we describe a different experiment aimed at comparing multi-touch and free-hand interfaces. Nowadays, various cheap techniques can be used to easily create different settings supporting a more natural interaction. Following this practice, we decided to develop an engaging virtual planetarium application in order to continue what we learnt about 3D gestural interaction in Chapter 7. Thanks to the work done at that stage, we can easily design and develop an environment specifically devised for the comparison of different gestural interfaces, in particular multi-touch and free-hand ones. We discuss the user tests carried out and finally we can draw some conclusions about the evaluation, analyzing the control and the users' involvement in the virtual environment.

## 11.1 Evaluation of gestures in multi-touch interaction

Gestures represent an easy to observe virtuous practice that in desktop computing appear limited almost exclusively to pointing with hand or finger, while observing users of multi-touch tables it often happens to see fluent, dual-handed metaphorical gestures. This raises the questions we try to answer. Is there any practical advantage (e.g., in terms of efficient problem solving) when using a natural interface? More precisely: is multi-touch better than the desktop for some traditional application? Moreover: can gesticulation be used as a suitable signal of natural interaction justifying the claim that more gestures provoke a more natural, and, thus, better interaction?

We opted to experiment with *pair-programming* [130]. It is a practice of software engineering strongly recommended by *agile methodologies* and, thus, represents a realistic and non artificial test-bed both for desktop and multi-touch settings. Additionally, gesticulation, which we aim to observe, is more easily, though not exclusively, triggered during group-work.

### 11.1.1 A short recap and motivations

As already discussed, multi-touch interaction has been a topic of research since the mid-eighties (e.g. [22,76,83,87]), but it's with the recent work of Han [55,56] that this interaction paradigm has become popular and multi-touch interaction is now so often taken as an example of natural interface.



However, applications based on this interaction paradigm are still in a phase of creative envisioning (e.g. [57, 133, 138]) and little, if any, study exists on the real advantages of direct manipulation in traditional application fields.

For example, Owen and colleagues [103] explore the advantage of bi-manual input on a curve matching task; Patten and Ishii [104] present a study that compares the strategies (and effectiveness) of spatial organization with tangible and traditional user interfaces.

These studies let foresee an advantage of direct manipulation, and by extension of multi-touch tables, over traditional desktop for very specific tasks that have in common a certain physicality, but don't settle the point on whether or not surface computing can replace the desktop in traditional work or learning scenarios.

Our goal is determining whether in multi-touch environments there is a significant increase of non-verbal communication in general and particularly, of gestures, during users performance in the task of understanding and debugging algorithms.

### 11.1.2 Comparing multi-touch and desktop interfaces

As previously highlighted, one feature of the multi-touch devices is the possibility to support collaborative tasks. If a device is large enough, it can lodge multiple people at one time, providing a place where user can give their own contribution and pursue objectives that are personal or in common with others. This way, people appears more confident and relaxed when approaching touch-based environments, than when using desktop setups. With our experiment described here, we intend to investigate whether multi-touch is actually useful in the workplace.



Figure 11.1: People participating in the experiment: at the multi-touch (left) and at a traditional desktop (right)

A convenience sample of 44 people, aged 20-35, all students of computer science or ICT professionals, thus quite literate in computer programming, participated to this study. Working in pairs (see Figure 11.1), the testers were asked to review 7 snippets of C code (1 demo, and 6 exercises), each one containing a bug, and point out the bug to an assistant. The review of the code snippets was performed through a very simple interface implemented with the identical look and feel both for the desktop and the multi-touch environment.

The appearance of the graphical interface is shown in Figure 11.2; it consists of

- a square text-area that shows one snippet of code at a time. The snippets of code are short enough to fit the visible area, so no scrolling is ever needed and no scrollbars are thus provided;
- a small control panel with a timer, and buttons to jump to forward and backwards between the exercises; multi-touch functionalities where enabled on the MT table and simulated with keyboard/mouse combinations on the desktop.

However, in practice, testers seldom manipulated the interface, except for hitting the *Next* button.



Figure 11.2: The appearance of the user interface

## Pre-test briefing

Before the beginning of a test session the testers were briefed on the purpose and method of the research. We had great care to specify that the goal of the work was to evaluate the quality of the tool (Desktop vs Multi-touch) and not the ability of the users. The need of a video recording was justified by explaining our need to monitor “collaboration and non-verbal communication”

but without explicitly mentioning gestures, or their supposed connection with efficient problem solving. The testers were then encouraged to cooperate to the solution of the problems. The testers were also informed that:

1. every snippet contains one (and only one) bug;
2. the bug is not in the syntax, but in the logic of the code;
3. comments (where provided) are not misleading;
4. bugs, although trivial to explain, were sometimes well concealed, and intended to be difficult to spot;
5. finally, although no time constraints were given, the testers were informed that the whole test required between 15 and 25 minutes on average. This was not intended to fix a goal for the performance, but to prepare the testers to the effort needed to complete the test.

Of course all participants were given written warranty of privacy and non-disclosure of videos and disaggregated data.

### **Test Session**

The 44 testers (spontaneously organized in 22 couples) were then asked to complete the experiment. 11 tests were run at the Desktop and 11 were run at the Multi-touch table, the assignment to one or the other setting was performed randomly. The F-test was used to verify if a significant difference exists between the two methods, multi-touch and desktop. Note that the same 7 exercises were administered at the 2 settings.

Of the 7 snippets of C code, the first one was intended as a demonstration to get into confidence with the interface and clarify latest doubts; results are not taken into account in the following discussion. For each one of the remaining 6 snippets, the testers had to perform the following:

1. examine the snippet for as much time as needed, discussing, if necessary, to decide what the bug was;
2. as soon as an agreement was reached on the exercise, press a pushbutton (that turns green) on the control panel;
3. testers could then point out the bug to an assistant, who annotated it in a block notes, without either confirming or refusing the answer;
4. by pressing a pushbutton on the control panel the testers could then proceed to the following exercise.

Note that in both settings:

- the interface didn't allow any editing of the C code; so the users were not able to correct the error;
- since the assistant did not comment on the proposed solution, the test actually measures the time spent before reaching an agreement, we did not measure the accuracy (i.e. if the testers positively solve the exercise or not) of the exercise; thus, wherever in the rest of the paper we talk of solving an exercise it should be clear that we mean reaching an agreement on the solution;
- the cases in which the testers were not able to reach an agreement (either on the correct or on a wrong answer), were also included; in a sense this results indicate the time spent before deciding that additional tools/information was needed to positively solve the exercise; of course such cases should better be taken into account in a deeper investigation;
- the testers hit a button after reaching an agreement and another one to switch to next exercise, thus the time spent in reporting the bug to the assistant is known and has been expunged in the following discussion.

## The 6 Code Snippets

The various exercises have been designed to be of increasing complexity and length (and in general took increasing time to solve). The exercise can be divided in 4 categories, and were administered in the same order in which they are described below:

**Type 1:** controversial exercises such as the one below are likely to cause debate between the testers.

```

1 void test2() {
2   int i;
3   for (i=0; i<10; i=i+1)
4     if (i=2)
5       printf("i is 2\n");
6     else
7       printf("i is not 2\n");
8 }
```

In the specific case the use of an assignment as argument of a truth evaluation, though not syntactically wrong, is typically deprecated. There are exceptions however, and the testers spent time discussing whether or not the use of such construct was acceptable in the context of the exercise.

**Type 2:** slips or careless errors are very common in everyday programming and are easily spotted since often result in meaningless or inconsistent code.

```
1 void test3() {
2   int i;
3   i = 0;
4   while (i < 10);
5     i = i + 1;
6   printf("Finished. i = %d\n",i);
7 }
```

In this case the body of the while construct is actually an empty statement (because of the semicolon), resulting in an infinite loop.

**Type 3:** pattern matching error are those ones that require visual memory or recognition, and represent a class of errors almost unknown to programmers today, thanks to the use of visual editors that provide syntax highlighting. Examples include misplaced parentheses due to a wrong indentation, and comments opened and not closed, such as in the example below.

```
1 void test4() {
2   int i;
3   for (i=0; i<10; i=i+1)
4     /* check the value of i */
5     switch(i){
6       /* is i 0? */
7       case 0: printf("i is 0\n");
8         break;
9       /* is i 1?
10      case 1: printf("i is 1\n");
11         break;
12      /* now the default case */
13      default: printf("i is more than 1\n");
14    }
15 }
```

Most modern editors would help the programmer to find the error here: the comment at line 9 is not closed at the end of the line, and runs through to line 12, voiding in practice the body of the function. Without the help of syntax highlighting, the testers were forced to check the syntax of comments, which is trivial in practice, but not intuitive.

**Type 4:** algorithm understanding exercises are those ones for which the most effort was required. The bugs consisted in the overrun of array indexes, such as in the example below.

```
1 void bubble_sort(int array[], int n) {
2   int i, j;
3   // sort array of length n
```

```
4  for (i = (n - 1); i > 1; i++) {
5      for (j = 0; j < i; j++) {
6          if (array[j] > array[j + 1]) {
7              // swap values
8              int tmp = array[j];
9              array[j] = array[j + 1];
10             array[j + 1] = tmp;
11         }
12     }
13 }
14 }
```

Here the outer for cycle will never end and causes an array overrun on the subsequent instructions. Testers were able to solve the exercise only after understanding (or recollecting from previous study) the basic logic of the algorithm.

## Data Collection

Data collected during or following the test are:

1. the time spent on each exercise;
2. the proposed solution, that may or may not be correct;
3. the video footage of the whole session.

These were used in the analysis described in the next section. Other data gathered, but not discussed in detail here are:

1. whether or not the testers were able to reach an agreement on the solution of the exercise;
2. subjective scores of the difficulty of each exercise.

This information, as we already noticed, will be subject to further investigation on the accuracy of the performance and on the subjective perceived difficulty of the exercises in the two settings.

## Analysis of the Video Log

To better understand the role of gestures in collaborative work we have analysed the video logs of the test sessions in order to count the gestural events. There is strong evidence that a fluent gesticulation has a positive influence, among others, on short term memory [50] and learning [31].

Our hypothesis is that a similar relationship can exist in solving complex tasks such as the one considered here, and that a system that allows (or encourages) a fluent gesticulation can lead to better performances. The video collected were annotated using Anvil [79], a platform for multi-layered annotation of video with gesture, posture, and discourse information.

## Experimental Hypotheses

As mentioned earlier, among the scopes of this work, a main goal is to answer the following research questions:

1. Is there any practical advantage (e.g., in terms of efficient problem solving) when using a natural interface? More precisely: is multi-touch better than the desktop for some traditional application?
2. Can gesticulation be used as a suitable signal of natural interaction (i.e., the more gestures, the more natural, and the better interaction)?

Hence the null hypothesis related to question 1.

**H1.** Participants will be no faster in solving an exercise containing a controversial bug, when using the Multi-touch table or the Desktop.

**H2.** Participants will be no faster in solving an exercise containing a careless error, when using the Multi-touch table or the Desktop.

**H3.** Participants will be no faster in solving an exercise requiring a pattern matching, when using the Multi-touch table or the Desktop

**H4.** Participants will be no faster in solving an exercise that require algorithm understanding, when using the Multi-touch table or the Desktop.

In order to positively answer question 2 we should first prove that the observed difference in fluency of gestures couldn't be otherwise explained:

**H5.** Participants will gesture with no more of less fluency (measured as gestural units/time) at the Desktop or at the Multi-touch table.

Further hypotheses, showing if fluency of gestures has any direct impact on efficient problem solving (i.e., couples with more fluent gesture actually perform better), or a deeper exploration in the nature of gestures involved in this specific task (e.g., what pantomimes, icons, metaphors were used in addition to deictics that helped the participants who scored the better results) are outside the scope of this work.

### 11.1.3 Results and Discussion

As shown in figures 11.3 and 11.4, the experiments prove that, for the task examined, people perform significantly better at the multi-touch table than at

the desktop for some of the exercises, namely those ones involving cooperation, discussion, and, generally speaking, exchange of communicational information.

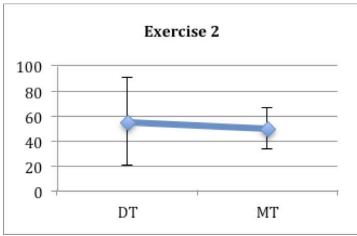


Figure 11.3: Results of exercise 2 (controversial bugs)

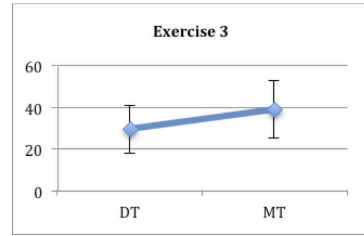


Figure 11.4: Results of exercise 3 (careless errors)

Do participants perform better when solving an exercise containing a controversial bug, when using the Multi-touch table than the Desktop? As shown in figure 11.3 tester scored slightly better performances at the Multi-touch; the difference is significant,  $F(10, 10) = 4.72$ . Hypothesis H1 should then be rejected. The analysis of results of exercise 3 does not show any significant difference between the Desktop and the Multi-touch,  $F(10, 10) = 1.46$ , n.s. figure 11.4 shows means and standard errors for the results of the experiments. Similarly, no significant difference was observed in the execution of exercises 4 and 5, both containing errors requiring a pattern matching: precisely:  $F(10, 10) = 1.29$ , n.s. for exercise 4 and  $F(10, 10) = 1.08$  for exercise 5. Figure 11.5 shows the results. Finally, exercise 6 and 7 required the most effort from the testers (as shown by the longer time to solve on average, figure 11.6), and the Multi-touch setting allowed a tighter cooperation resulting in a significant better performance:  $F(10, 10) = 5.56$  for exercise 6,  $F(10, 10) = 13.50$  for exercise 7. The timing are summarized in table 11.1.

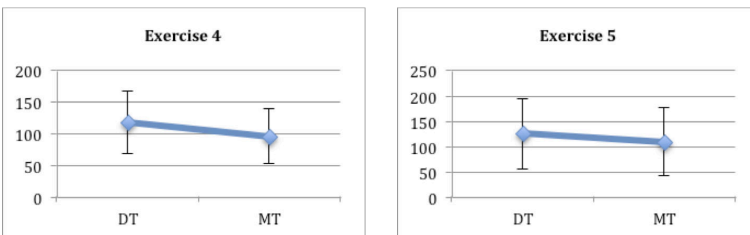


Figure 11.5: Results of exercise 4 and 5 (pattern matching)



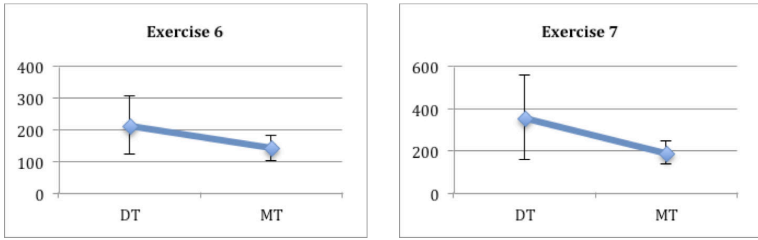


Figure 11.6: Results of exercise 6 and 7 (algorithm understanding)

Exercise #	Avg. Time (Desktop)	Avg. Time (Multi-touch)
2	55.18	49.91
3	29.64	39.18
4	119.27	95.82
5	125.64	109.36
6	213.55	143.09
7	356.55	190.80

Table 11.1: Time spent on average on each exercise. Desktop (left column) and Multi-touch (right column) performances are compared.

### 11.1.4 Gesture Fluency

Our last test was aimed at showing if users manifested a difference behaviour with respect to gesture fluency in the MT and DT settings. We observed proper gestures according to the related literature given in Part 2. In particular:

- Only movements of the hands were counted as gestures, thus excluding nodding and changes in body postures; specifically, pointing with the mouse was not counted as gesturing; in fact, mouse pointing is not a proper gesture and comparison to previous work is problematic. Additionally, we can't assume the visibility of the mouse gesture to the other user, i.e. there is no clear communicative intent (see later).
- Movements of the hands were counted as gestures when they had a clear communicative intent: folding the hands together is not a gesture; pointing, mimicking an action, and counting with fingers are all considered gestures;
- Gesture phrases were counted as their atomic components where possible; for instance, when a tester points a section of code, then another to show correlation, and finally makes sharp movements to show progress, even if these three movements are executed without any visible pause, were counted as 3 separate gestures.

We, thus, introduced for simplicity a measure of gesture fluency, as the number of gestural events per second of both testers, and, for each one of the 22 couples, we counted the gesture events of both testers. The gesture fluency of a couple is the total number of gestures performed divided by the total time spent solving the 7 exercises. Results are shown in figure 11.7. The experiment shows that

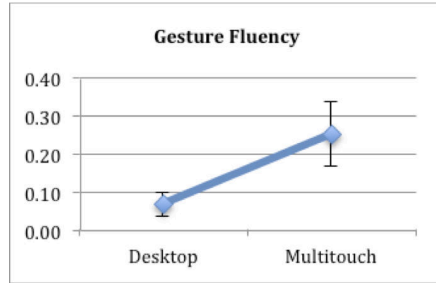


Figure 11.7: Gesture fluency compared for the two settings: desktop (left) and multi-touch (right)

participants use significantly more gestures when using the multi-touch than when interacting at the Desktop,  $F(10, 10) = 7.70$ ; thus we reject hypothesis H5.

### 11.1.5 Summarising

As we only informally introduced before, the results shown above indicate that while working at the multitouch table people perform better than at a traditional desktop, and such improvement is associated to an increased gesticulation.

Some remarks however are due here. Not all types of exercise seem to benefit from the adoption of a multi-touch system, in particular snippets containing careless errors (exercise 3) and pattern matching (exercises 4 and 5) were not significantly affected by the different setting. Controversial exercises (such as exercise 2) were better addressed at the multi-touch, where a tighter cooperation was possible. This is hardly a surprise, since this sort of problems requires discussion and sometimes negotiation between the users.

The results obtained for exercises 6 and 7 (algorithm understanding) are perhaps less intuitive and their implications in the design of interactive applications deserve some attention. On the one hand this result is coherent with the already observed connections between gesturing and problem solving. In this case an improvement in the graphical interactive systems did not involve improvements in the interface (as remarked above, people didn't do extensive use of the multi-touch features of the tabletop setting), but rather the design of

a work-setting more suitable for cooperation, and fluent gesturing was taken as a metric for the cooperation itself.

On the other hand, one can notice that the exercises taking the most benefit from the multi-touch setting were the more difficult among the 7 administered, and still were trivial with respect to the typical problems that programmers face in daily work. Our experiments suggest that multi-touch tables, encouraging cooperation, help people express their potential, thus resulting in a better performance.

The registered difference in performances for code understanding and debugging time could make a significant difference in many practical cases. If confirmed, these results may help reconsidering the design of our offices and programming labs towards a more widespread adoption of tabletops, that today are mostly regarded as research prototypes and curiosities.

Some further empirical observations are worth mentioning here. Our metric of gesture fluency was suitable for the work at hand, but hides the real complexity of gesture phrases. If the gesture largely more exploited by all participants was pointing with one finger, others were frequently observed:

- Gestures indicating progress or continuity, both single and dual-handed, are executed moving the hand(s) on a circle or sharply from left to right; such gestures are not easily performed when sitting, and not surprisingly they are less frequently seen at the Desktop;
- Some gestures are performed primarily for communicating, they are a sort of visible words; as such they have to be performed in a well defined and visible space; again, such space (close to the screen) is easier to reach at the multi-touch than it is at the Desktop;
- At the multi-touch pointing with the finger was sometimes used to negotiate the attention of the mate; testers often pointed at the same point on screen as to reinforce and confirm a gesture; this behaviour was not observed at the Desktop;
- In one case a tester asked if she could use paper and a pencil, which was not possible, actually; several participants at the multi-touch setting were observed while mimicking the use of paper and pencil on the palm of the open hand.

However, as noted throughout the work, several questions remain open.

Firstly, we didn't observe the accuracy of the solutions proposed. For the purpose of this research a problem was considered solved when an agreement on the proposed solution was reached; though in principle co-operation and discussion lead to accurate results, a precise measure of such accuracy is likely to expose new insights.

The choice of observed gestures was arbitrary, though shared in literature; for example, pointing with the mouse is a common behavior at the Desktop, and its impact should be evaluated.

How strong is the interrelation between gestures and efficiency/accuracy? Do couples that show more gesture fluency perform better?

Finally, new insights may come from a more detailed analysis of gestures; gesture fluency doesn't capture the richness of expression that emerges at the multitouch table, where dualhanded symbolic gestures are often used compared to bare single-handed deictic that form the majority of gestures at the Desktop.

## 11.2 Comparison between multi-touch and free-hand interaction

As described in previous chapters, even low-cost sensing devices can provide engaging interaction experiences, trying to put into practice the ubiquitous computing vision, where the user is expected to interact naturally with the technology without even realizing its mediation.

This way, designing engaging interactive environments and then using them for the evaluation and comparison of different interaction paradigms, it is quite simple.

In this section, we report an experiment on the 3D exploration of a virtual planetarium in order to compare two different interfaces. The first interface is based on a simple multi-touch paradigm, while the second one exploits a free-hand interaction together with a projection on a geodetic sphere. We describe the technical implementation of both versions in detail. Then, we discuss the results of user-study comparing the two modalities, and highlighting a tradeoff between the control and the users' involvement in the virtual environment.

### 11.2.1 Controlling a planetarium application

Nowadays, the availability and the decreasing cost of various sensing devices allows for the creation of interactive spaces, especially in public and shared settings, even with limited resources. We describe how we followed this philosophy to create a more immersive and engaging version of an existing virtual planetarium software for desktop systems, transforming it into both a multi-touch and a free-hand application. In particular, the free-hand version exploits also a geodetic projection on a hemisphere, enhancing the realism of sky visualization.

## 11.2.2 Immersive Systems

Many investigators used the sky and space exploration to provide examples of immersive systems. A description of how virtual environments can be exploited for such kind of tasks can be found in [101] which describes, among the other settings, how such a kind of environment is exploited by the NASA. Another relevant example is the work in [117], where the authors exploited magnetic sensors in order to support the user while pointing or searching for real stars. In addition, they exploited also the Wiimote controller for guiding the recognition of the constellation shapes.

In [5], the authors exploited a spherical display for creating a 360-degrees space for visualizing content for multiple users minimizing the occlusion. The authors exploited such display for collaborative settings. We used a larger spherical display, and we implemented a free-hand interaction paradigm with such screen. In addition, we projected the image on the concave surface rather than the convex one.

Our setting is characterized by the same 3D interaction technique proposed before (Chapter 7), adding a more realistic projection of the sky map, in order to increase even more the user's immersion feeling.

## 11.2.3 Prototype design

We discuss now the two different interaction settings we created for an interaction-enhanced version of a planetarium virtual environment. We used the Stellarium [63] software, which is able to show a 3D view of stars and planets according to the current time and the user-specified position in the world.

The application lets the user browse the sky using the mouse and the keyboard for moving the current point of view, and/or selecting planets and stars. The basic version of the application is controlled using a two button 2D (or 3D) mouse: moving the position of the pointer moves the current view, the left button can be pressed for rotating or panning the view, while the right one is used for scaling. However, this standard scheme of scene exploration can be easily enhanced for offering a more engaging environment. In order to support a more natural interactive scheme, we designed an interactive model where users perform their tasks in a three dimensional space or touching directly the sky representation.

The setup described in detail in Chapter 7 is exploited here to build both versions, giving us the chance to develop a reliable multi-touch interface and a complete gestural free-hand interface. In both versions, we used the same technical solution for implementing the communication between the two different gesture recognition sensors and the existing Stellarium application. Following our early experience with free-hand interfaces, the communication is based on TUIO [75] network protocol. On one hand, we created an intermediate layer

between the devices and the application that generates TUIO events according to the current state of the device's tracking; on the other hand, we extended the Stellarium code which allowed us to translate such events into application commands, like rotating or panning the view or moving the current camera position.

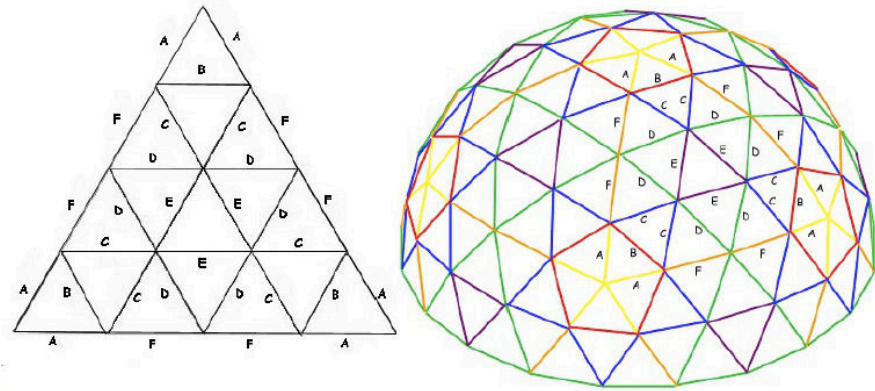


Figure 11.8: Geodetic sphere scheme.

### Multi-touch version

We used the multi-touch table exploiting the improved FTIR table proposed in section 6.1. The resulting interaction setting is shown in Figure 11.9(a): the sky visualization is projected onto the multi-touch screen. The user controls a rotation hinged on the barycenter of the scene by touching the scene and moving the finger in different directions. In addition, it is possible to resize the scene touching the surface with two fingers: moving them apart from each other enlarges the scene, while moving the finger towards each other shrinks the scene proportionally.

With respect to the desktop version, the multi-touch interface has the obvious advantage of allowing the user to interact directly with the sky projection, without mapping the mouse movement into the scene. In this work, we consider such setting as the baseline for a low-cost solution for creating an application deployment that can be exploited in a shared and/or public setting.

### Full-body version

We enhanced the virtual planetarium experience with a different environment, which mimics the visualization of the sky ceiling through the projection of the sky map on a hemispherical surface.

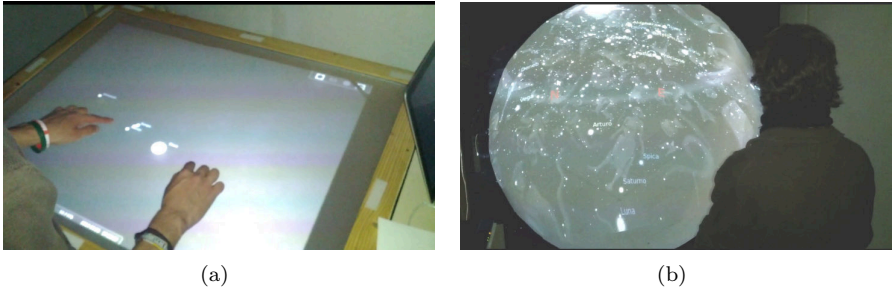


Figure 11.9: a) The multi-touch planetarium application. b) Full-body version of the planetarium application projected on a geodesic sphere.

In order to keep low the cost of the implementation, we have built the entire surface using paper and glue. Following the scheme in Figure 11.8 we built a hemispheric mesh of kraft paper triangles. After that, using thin wood planks we built the shell holding the surface. The shell length and height are equals to the diameter of the hemisphere, which is about two meters. After securing the hemisphere to the shell, we painted the surface using a common white wall paint and finally we placed a black cover between the squared shell and the hemisphere borders which has a circular section. Once completed the construction of the hemisphere, we used a short throw ratio projector for displaying the sky map on the curve surface. In this case, the projection exploits an orthographic filter that is provided with Stellarium. The resulting setting is shown in Figure 11.9(b).

The interaction with this a kind of screen is based on a free-hand paradigm, using a Microsoft Kinect sensor. Such a gestural interaction makes it possible for the user to navigate the sky map. Thanks to the improved free-hand interface described in section 7.3.2, we exploited a grab gesture that has to be performed by the user to set in motion the interaction with the planetarium application. Therefore, in order to start the interaction with the application, the user has to close at least one hand, which resembles the act of grasping a real object.

The user can interact with the sky visualization rotating the scene around its barycenter and enlarging or reducing it in order to get the desired level of detail.

We support the interaction through two gestures:

- rotation is supported simply performing an on-air grab with either the right or the left hand. Keeping the hand closed the user changes the hand position, and the view is rotated accordingly;
- the zoom is supported through a two hand gesture, closing both hands. The user can either enlarge or reduce the view size respectively moving

the hands apart or moving them towards each other.

After building the projection screen and implementing the interaction gestures, we had our interaction scenario ready: the user was able to see the projection of the planetarium image on a spherical screen and to interact with the application simply standing in front of the hemisphere and performing free-hand gestures. Such setting is engaging not only for a single user, but also for a group of people: one of them controls the visualization while the others look at the projected sky map.

### 11.2.4 Evaluation

We conducted a small-scale user test in order to compare the two interfaces for the interactive planetarium, which gave us some hints on the differences encountered by users in the different platforms (namely multi-touch and Kinect).

The aim of the proposed test is threefold. First, we wanted to evaluate the overall perceived difficulty while executing different tasks with the planetarium interface in both settings. Second, we wanted to determine whether the overall usability of the application was affected by the change of platform. Finally, investigated if there were differences in the factors that affect the user's presence perception, according to [134].

The test was organised as follows: after completing a small demographic questionnaire, the users had to complete the following tasks using both versions of the application:

1. Starting from a visualization where Saturn occupies the whole screen, the user had to go back to a point of view where it is possible to see the Earth.
2. The user had to complete a 360° horizontal rotation of the view (on the Z axis).
3. The user had to find Jupiter and visualize the name of its satellites.
4. The user had to change point of view in order to see at the center of the screen one constellation (selected by the user, but declared at the beginning of the task)
5. Starting from the a view of Saturn with a minimum zoom factor, the user had to enlarge it until it occupied the whole available space.

We alternated the starting version in order to minimize the carry-over effect. After the completion of each task, the users answered the Subject Mental Effort Question (SMEQ) [139] in order to evaluate the perceived difficulty. After completing the whole task set, the user had to fill two different questionnaires: the first one is the Software Usability Scale (SUS) [14] that evaluates the overall



usability of the application, while the second one is the Presence Questionnaire [134], which measures different aspects of the user's presence perception. After completing both questionnaires, the user repeated the experiment with the other version (multi-touch if starting with the free-hand and vice-versa).

Thirteen users participated to the test, 9 males and 4 females, aged between 21 and 26 ( $\bar{x} = 23.3, \sigma = 1.8$ ), 5 had a high school, 5 a bachelor and 3 a master degree. The users were more proficient with multi-touch applications ( $\bar{x} = 5.54, \sigma = 1.6$  in a 1-7 Likert scale), if compared with the free-hand one ( $\bar{x} = 4.3, \sigma = 2.1$ ).

For the post-task evaluation, we report in Table 11.2 the upper bound ( $\rho = 0.05$ ) of the perceived user effort. According to [14], the perceived effort for all tasks is between 11 and 25, labelled respectively "*Not very hard to do*" and "*A bit hard to do*". It is possible to notice that the multi-touch version required less effort for T1 and T2, while the free-hand version performed better for T4 and T5.

Task	Multi-touch	Free-hand
T1	8.25	15.50
T2	8.67	13.80
T3	7.40	10.00
T4	18.76	11.96
T5	17.21	10.00

Table 11.2: Perceived task difficulty upper bounds ( $\rho = 0.05$ ).

The SUS post-study questionnaire did not revealed any difference in the overall usability of the two versions. The score was  $\bar{x} = 74.04, \sigma = 11.67$  for multi-touch and  $\bar{x} = 70.97, \sigma = 11.57$  for the free-hand version. Therefore we can conclude that the perceived usability of the two versions is about the same.

For the presence post-study questionnaire, we disaggregated the scores of the different answers (1-7 Likert scale) according to the following factors [134]: Control Factors (CF), Sensitivity Factors (SF), Distraction Factors (DF) and Reality Factors (RF). Obviously, given the small number of participants, it is not possible to generalize the quantitative results. However, we want to point out here a qualitative tendency that explains the different perception of the effort for the different tasks.

In Table 11.3 we report the questionnaire results. The multi-touch version performed slightly better for the CF and the DF, while it was slightly worst for SF and RF. This means that the users had more difficulties with the free-hand version when a fine-grained control of the planetarium positioning was required (T1 and T2). However, the users were more involved from a sensory point of view, and they found more real the free-hand experience. Indeed, the more exploratory tasks had a higher rating with the free-hand version (T4 and T5).

Factor	Multi-touch	Free-hand
CF	$\bar{x} = 4.96, \sigma = 0.82$	$\bar{x} = 4.77, \sigma = 0.22$
DF	$\bar{x} = 5.19, \sigma = 0.49$	$\bar{x} = 4.92, \sigma = 2.12$
SF	$\bar{x} = 5.15, \sigma = 1.2$	$\bar{x} = 5.5, \sigma = 0.22$
RF	$\bar{x} = 3.62, \sigma = 1.85$	$\bar{x} = 4.04, \sigma = 1.63$

Table 11.3: Disaggregated results of the post-study presence questionnaire.

From these results we can conclude that, given a comparable overall usability and cost of the two settings, it is better to select the multi-touch environment for a more fine-grained control, while if we want to increase the sensory and realism perception for the user (according to definitions in [134]), it is better to select the full-body version.

### 11.2.5 Summarising

In this section, we compared multi-touch and free-hand environments in the exploration of a virtual planetarium. For this purpose we designed a cheap setting for an full immersion planetarium experience. The starting point was an existing software for desktop platforms and we created both a multi-touch and free-hand version of the application. The free-hand version employs a geodetic display that gives the user a more accurate representation of the sky.

We performed a small-scale user study in order to investigate the perceived difficulty in performing different tasks with the two settings, which was quite modest for both versions. In addition, the post-test questionnaires did not highlight any significant difference in the overall usability between the multi-touch and the full-body interaction.

However, we found a difference in the perceived control of the application (which was higher in the multi-touch version) and in the perceived realism of the experience (which was higher in the free-hand version).

**Acknowledgement:** This chapter is based on the papers:

Alessandro Soro, Samuel Aldo Iacolina, Riccardo Scateni, and Selene Uras. 2011. *Evaluation of user gestures in multi-touch interaction: a case study in pair-programming*. In Proceedings of the 13th international conference on multimodal interfaces (ICMI '11). ACM, New York, NY, USA, 161-168. [119]

Some of the C code snippets were adapted from fragments available online under GNU GPL or analogous licenses. Snippets 1-4 were adapted from [89].

E. Tuveri, S. A. Iacolina, F. Sorrentino, D. Spano, R. Scateni. *Controlling a planetarium software with a Kinect or in a multi-touch table: a comparison*. Proceedings of CHIItaly '13: ACM SIGCHI Italian Chapter International

---

Conference on Computer-Human Interaction, p.6, September 2013, Trento, Italy. [124]



## Chapter 12

# Case study: Architecture and Construction

Once we evaluated and tested our interfaces, we applied a more gestural paradigm, originated by the technological improvement already described, to applications in the field of Architecture and Construction. Part of this doctorate work has been conducted in a company<sup>1</sup> that deals with topographic surveys, development and commercial distribution of software for tridimensional data processing.

In this context the objective is to find innovative solutions that would allow the natural exploration of tridimensional contents (representing buildings, terrains, squares and so on) facilitating the work of engineers and technicians and attracting casual users to the Architecture world.

Firstly, we see the motivations and reasons that led us to make such a choice. Since human-computer interaction is based on human senses and the vision is considered the most predominant sense, we have to develop an efficient rendering system allowing the visualization of high resolution 3D models, preserving the data high quality at the same time. Then, we focus on how we can develop more manipulative and gestural interfaces with the intent of creating suitable working environments specifically devised for this area.

### 12.1 Background

In Architecture and Civil Engineering, digital models are increasingly replacing paper projects. What was once done by hand like the drawing of a plan or a map, is now replaced by a set of automatic and semi-automatic steps. Thanks to recent improvements in Lidar technology, 3D Laser Scanners that measure large

---

<sup>1</sup>Gexcel S.r.l, University of Brescia spin-off

scale-objects, such as entire buildings or squares, with high accuracy and high resolution, make realistic and extremely detailed 3D models. Moreover the high performance offered by parallel computing platforms allows the data extraction from these digital models. Algorithms and software techniques, for instance, can be used in order to analyze raw models and produce highly detailed vectorial data.

The use of high resolution 3D models, however, increases the difficulty of interactive tasks in a uncontrollable way, requiring specialized skills to manage this large amount of data. The exploration of large models becomes more wearing, and standard HCI instruments fail to provide a suitable working environment in terms of hardware devices/tools and software interfaces. The aim of this chapter is the description of innovative exploration techniques and high resolution 3D models navigation, in order to contain the interactive task's load and, preserving the data high quality at the same time.

## 12.2 Interacting with massive point clouds

The most accurate measuring instrument in the civil engineering field is the laser scanner. This device scans a real-world object or environment storing information on shape, appearance and color, at least in grey scale. The output consists in the so-called 'point cloud', or 'cloud', a set of points in 3D space. This type of data is still raw and discrete but with the help of some algorithms we can extract a mesh, a mathematical representation of the surface that provides an approximated shape. The higher the resolution of a laser scanner, the more accurate the acquired shape, and the closer is the calculated surfaces to a physical object.

The cloud is therefore a virtual representation of real objects. Thanks to its measurable nature, it is possible to analyze the point cloud extracting specific properties. In Engineering users need to extract curves and silhouettes to draw vectorial representations with CAD applications. In addition, since the point cloud preserves the unity/unit of measure, it is possible to extract distances to produce detailed 'as-built' drawings. The phrase "as-built" in construction is equivalent to "as-is". Drawings deemed "as-built" are thus drawings that reflect/display the existing conditions as they are, or "as-is". As-built drawings are included in the standard documentation of construction projects, they can be documented either during or after construction. In both cases, a point cloud is therefore used to compare an existing CAD project, representing how the construction will be built, showing the actual real measures extracted from the scanned point cloud.

Laser scanners suitable for different scales can accurately acquire geometric and color properties with high precision, producing point clouds of billion of points. When technology reaches this level of precision/accuracy, the collected data tends to be extremely large in term of size on the hard disk.

Even latest generation computers are not adequate enough to the loading/s-

toring demand. GPUs are optimized for fast rendering of a huge set of three-dimensional primitives and high quality textures. However, the digital rendering of a territory or a valley may be so large that it can't be loaded in the graphic card memory therefore it can't be rendered as it is. Sub-sampling or cropping techniques are used in order to reduce data size, but this may cause loss of desired detail level.

Besides the hardware requirement, the management of massive clouds through classical interfaces implies a growth of the interactive load. Many CAD software applications provide specialized interfaces to create, render and edit 3D models. These interfaces are so complex that require technical abilities and a training period. What's more, if the user manipulates a massive 3D cloud in order to detect some objects and check their state, it's hard to keep everything under control. When working with data representing entire buildings, districts or city areas, the detection of items and the taking of measurements becomes a difficult task. The model comes with so many details that, the navigation in the 3D space tends to be slightly dispersive without proper tools, and it gets hard to stay focused on the parts that are of most interest.

An explicative example is the mobile mapping. In recent years Government Institutions are frequently using cars equipped with 3D scanning devices to retrieve detailed 3D representation of buildings and streets. By using specialized software applications, technicians analyze these gigantic 3D models to find specific objects, such as all the traffic lights or/and all the streets signs. Once an object is manually recognized and classified, some relevant information, like the type of sign, its height or its position in the city map, is stored in a database. This process helps institutions in creating more accurate city maps, in monitoring the state of the road surface and, once all road signs are recognized, in simulating the flow of vehicles in certain predefined situations helping in critical configurations. The manual process of recognizing an object is obviously a very important step and maybe the most susceptible to errors. Operators are forced to use some kind of application software to explore the 3D model going through the roads, moving forward in the model meter by meter, and select the objects the user wants to annotate and catalogue. An easily usable interface is highly recommended because it leverages efficient and effective navigation and the execution of repetitive annotation tasks. We need to work hard on the exploration of 3D models with the aim of providing a easy way to manipulate them.

## 12.3 Multi-resolution Solid Images

As explained in the previous paragraph, the management of massive 3D models can be very hard due to a series of problems. The exploration of a large point cloud is not trivial while the detection of objects and features and the taking of measurements in a virtual 3D space are difficult tasks for an untrained user. Moreover the management of a 3D model forces engineers to work with a

dedicated computer or workstation, which is difficult to take with you, especially in the construction site. Since a lot of features can be detected directly from a 2D image, usually technicians prefer to work with 3D point cloud as less as possible and use to extract a series of images from the 3D model.

Common 3D software application allow the creation of an image from a 3D model in just a few steps. The user has to move in a 3D virtual space in order to choose the correct view focusing on the portion of the 3D model that interests him/her. Then he/she chooses the type of projection, orthographic or perspective. The most used projection is the perspective one, because it preserves distances and angles. Finally the process of generating an image, a 'rendering', from the 3D model is started. A rendering is the starting point for civil engineers and architects, from which lots of information can be extracted. For example with manually tracing the door silhouette from a rendered image, or windows as well as facades, the users can create a drawing of the shape, which, at a later stage, can be used to draw a plan or a prospect using a CAD program. The image quality extraction during the rendering stage is very important: a high quality rendering image allows the extraction of a more detailed drawing of the silhouette.

In certain circumstances a simple image is no longer sufficient for a number of reasons. The main problem is that the rendered image loses the unit of measure, thus the image is still helpful to detect the proportions of the shape or the silhouette of objects and buildings, but it cannot be used any longer to check the actual physical solid shape of a real object. Another problem can be the detection of silhouettes and features from a colour image. Especially in buildings that have a uniform color it is very difficult to discriminate indentations and protrusions.

Our solution proposes to render a 3D model through a special image, called 'solid image' [1, 9], that preserves both the color channel and the depth information. In other words the solid image preserves the measurability of the 3D model from which it is generated. This way, the solid image can be very helpful in producing as-built drawings or the solid image can be considered as a as-built drawing itself. However, this type of data has, like any other digital image or like any 3D model, intrinsic size problems. Even last generation computer platforms can only manage images with limited resolution. We face this problem in everyday use our computer when browsing an image gallery: the system struggles in opening a very large image at once. To overcome this problem we developed a technique that exploits image filtering to build a multi-resolution image. Thanks to a tiled rendering, this technique optimizes the data accessing time during the navigation. Combining the navigation simplicity of 2D images with the expressive power of a 3D model, our solution allows the visualization of massive 3D model, preserving its high accuracy and resolution. In order to test the technique's flexibility, a web-based viewer was developed. Designed to work on a browser, it supports on-line navigation and it is also compatible with the most common multi touch devices, such as smart phones and tablets. An immediate practical consequence is that civil engineers can now work on the



construction site, away from their home or office, using their own mobile device. Describing our idea from a different view point, the aim consists in showing a portion of a 3D model as a 2D image. Since images are easier to manipulate, a simple interface allows the user to explore the image’s contents at different scales without requiring any specialized pre-learned skills.

### 12.3.1 Related research

In this section we describe the techniques we used to overcome all the previously mentioned problems. The solid image preserves the depth information of the 3D model from which it is extracted, while a multi-resolution approach allows the production of a high resolution solid image ensuring, at the same time, a very smooth and reactive exploration.

#### Solid Image

Usually every standard image is made by 3 (or 4) matrixes, containing information of the primary red, green and blue colors; it also includes eventually the opacity information of the single pixel, called Alpha channel.

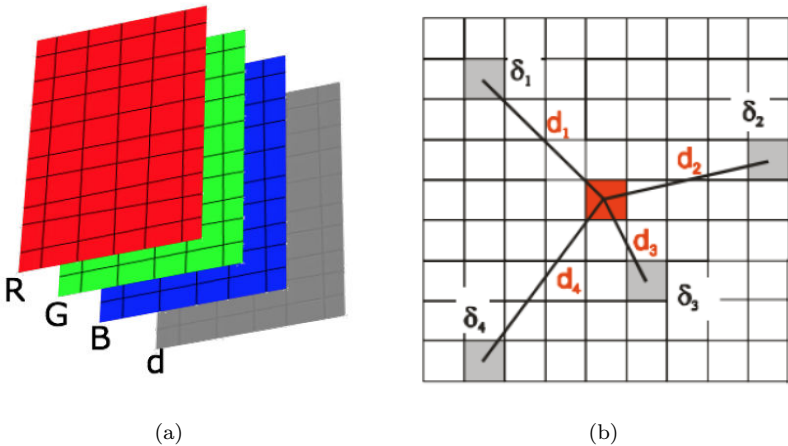


Figure 12.1: a) Structure of the solid image. b) Interpolation of the distance matrix.

A Solid Image [1,9], in addition to these data, adds another matrix called depth map. This matrix contains, for every pixel, the distance between the 3D point of the model that falls in the pixel and the centre of perspective. In other words, the extraction of a colour Solid Image, generally implies the use of: a simple image, obtained by means of a digital camera, providing the colour

channels; a laser scan containing geometric information in the three-dimensional space.

To extract the 3D information for every point, building the depth map, we need some data to correlate the image and the scanned point cloud. The calibration process, also known as alignment process, allows the alignment of photo and scanned image. This operation is usually done by placing some markers at known distances in the scene before scanning. These markers have a high reflectance property, so they can be easily detected both in the scan and in the picture. Joining the distance information, from marker to scanner, and from marker to marker, we can compute the view details with maximum precision. We can align the scan with the image, and, if we have enough markers, we can correct an optic aberration, especially the spherical distortion. In fact lasers scanners usually don't have any front lens, whereas a digital camera, has it. These aberrations are very common in architectural environments, because of the use of wide-angle lenses, notoriously more sensitive to this kind of problems. All these processes can be automated using ad hoc software algorithms [111,114]. The image usually has more pixels than the scan, therefore, the 3D-to-image direct mapping will show lots of holes where a 2D pixel does not have a corresponding 3D value, because in that region there was no survey. In order to overcome this problem, the holes are "filled" repeating the scan from other points of view, or with weighted interpolation techniques.

For example, if a single 3D point is surrounded by  $n$  pixels, the computation of the interpolated value of distance is carried out by the formula:

$$\delta_{i,j} = \frac{d_1 \cdot \delta_1 + d_2 \cdot \delta_2 + \dots + d_n \cdot \delta_n}{d_1 + d_2 + \dots + d_n}$$

where  $i, j$  are the indexes of the current pixel,  $\delta$  are the distance values to the object points, and  $d$  are the distances, on the image, between the pixel  $i, j$  and the pixels used for the interpolation. At this point, every single pixel of the image can be mapped to the corresponding model's 3D point.

Once we are done with all these operations, we can store the colour image and the depth file together and the Solid Image is built.

## Multi Resolution

In the real space the observer focuses on some details of the scene getting closer or farther to the subject. In a similar way, browsing a digital picture album, we change the zoom factor to enlarge the picture. The computing process that shows the image, however, works on the whole number of pixels, even if the size of the shown image is small. This is a waste of resources because many image pixels collapse in a single display pixel at rendering time. A multi-resolution approach allows the observer to examine the image features at many levels, almost like a real scene. This leads to a series of advantages, amongst others, a

minimized system stress.

A multi-resolution image stores some information thanks to which, with appropriate methods, it makes it possible to trace the image details on multiple resolution levels. The most common multi-resolution technique is the Gaussian pyramid [41].

**Gaussian Pyramid** The Gaussian operator is a smoothing/blurring filter used to remove the so called ‘Gaussian noise’, to transform the image in a much simpler one and to provide a more regular subsampling compared to other methods. With the Gaussian filter a multi-resolution image is created in 2 ways: if we create all lower levels from the original image, keeping the same size for all images, and increasing the kernel size at the decreasing of the level, we talk about space scale representation; otherwise, if the image size decreases when the kernel size increases, we talk about Gaussian pyramid. Each level of a Gaussian pyramid is half the size of its predecessor.

The construction of a Gaussian Pyramid is based on 2 fundamental operations

- Smoothing
- Downsampling

The first step consists in the application of a sequence of smoothing filters each of which has twice the radius of the previous one. So, if we call  $g_0$  the original image, the smoothed image is defined so that:

$$\hat{g}_1(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) \cdot g_0(i + m, j + n)$$

with  $w$  a 5x5 filter. Compressed formula is

$$\hat{g}_1 = w * g_0$$

The second level smoothed image will be:

$$g_2 = w * g_1 = w * (w * g_0) = (w * w) * g_0$$

The union of these two filters can be seen as one single filter

$$h = w * w$$

whose radius is twice than  $w$ . For level 3 the smoothed image will be:

$$g_3 = w * g_2 = w * (w * w) * g_0 = (w * w * w) * g_0$$

with a filter that has 4 times the size of  $w$ . And so on for all levels.

Once the image smoothing operation is made, we can go ahead and down-sample it. The final image will have a half the height and a half the width of the smoothed image. This step can be done without losing information because the smoothing filter reduced the apparent resolution. So, the downsampling operation will be the following:

$$\widehat{g}_1(i, j) = g_1(2i, 2j)$$

The two steps of smoothing and downsampling can be rewritten in a single function

$$g_k = REDUCE(g_{k-1})$$

which means for levels  $0 < k < N$  and nodes  $i, j$   $0 < i < C_k$ ,  $0 < j < R_k$  with  $R_k$  and  $C_k$  sizes of the  $k$ th level

$$\widehat{g}_k(i, j) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) \cdot g_{k-1}(2i + m, 2j + n)$$

### 12.3.2 The proposed solution

The main idea is basically to combine the previously described techniques: a Solid Image becomes a multi-resolution image by means of a tiled rendering. Based on *divide et impera* paradigm, the proposed approach builds the multi-resolution pyramid starting from several small images, called tiles, which represent portions of the original image. Following a bottom-up strategy the first step is to calculate the tiles number and their position. Then the tiles are created one by one, placing the camera in the previously computed positions. In the next step the matrix  $d$  of the distances between the point of view and the 3D model is extracted building the solid image for each tile. Finally all tiles are arranged to build the entire image.

#### Construction

We can easily compare the multi-resolution pyramid construction as a quadtree, where each leaf is associated to a different tile (figure 12.2(a)). In the complete tree the bottom layer has number of leaves =  $4^{lev}$  representing the full resolution

image (see the right side of the figure 12.2(a)). On the other hand, the top layer having  $lev = 0$  with  $width = height = 1$  represents the smaller image (the left side of the figure 12.2(a)). Other middle levels are composed so that  $4^{lev-1} < max(width, height) \leq 4^{lev}$ .

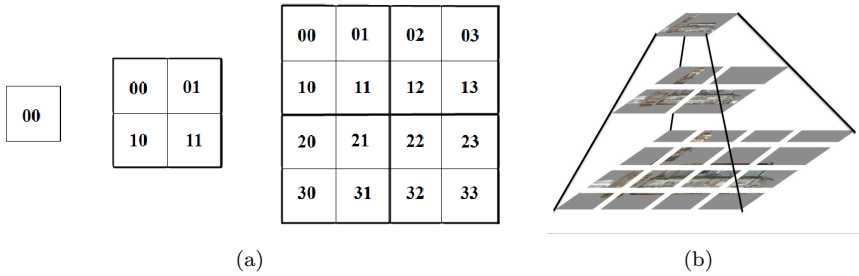


Figure 12.2: a) Example of the first 3 levels of the quadtree. The brother nodes are surrounded by a thick line. b) The multi-resolution pyramid.

If, for instance, the original image has a basic size of 8192 by 8192 pixels and the tile has dimension of 512x512, then the associated multi-resolution image set may contain a series of 5 layers, each one-fourth the total area of the previous one: 8192x8192, 4096x4096, 2048x2048, 1024x1024, 512x512 (the top layer), each containing respectively 256, 64, 16, 4 and 1 tiles.

The easiest way to create the bottom layer is to choose an appropriate number of rows and columns to split the image, having exactly  $4^{lev}$  tiles, in the combination of  $2^{lev}$  rows, and  $2^{lev}$  columns. A leaf can have square or rectangular shape and their number is usually a power of 2 or 4. The tile size must be the same for every tile in each level, but theoretically height and width can be different. With this combination, we assign the Top Left tile to the 0-0 leaf. In the bottom layer we take into account not only the colour image, but also the depth matrix and a matrix containing the concatenation of the 3 camera matrixes, the view, the projection, and the world. The visualization algorithm uses this information to retrieve three dimensional data corresponding to each pixel. The calculation of the tiles number and position starts the rendering process: the algorithm stores a colour image and a depth matrix as a float array for each tile. In other words, to create the bottom and most detailed level, the algorithm shoots all tiles one by one using an off-screen frame buffer object (FBO) provided by OpenGL: the information about the color is stored into an image, the depth data are redirected from the OpenGL depth buffer into a float array.

As a bottom-up approach suggests, the next levels of the pyramid are built merging the leaves 4-by-4 up to the top level. Four adjacent tiles, placed as a  $2 \times 2$  matrix, are merged together in a single image. This image, originally having a resolution equal to four times the tile's resolution, is downsized to the normal tile size through a gaussian bilinear filtering. Once the algorithm has

built all images in a level, the same routine is applied to the next one. This way the pyramid is made of images of the same size and its top level has only got a single image (figure 12.2(b)).

Although the algorithm builds all the colour images, to avoid data redundancy, the depth data are stored only for the bottom level. We opted for this approach because, the viewer shows typical 2D image during navigation, as the reader will see in the following section, On the other hand the 3D coordinates are only displayed on user's request usually for a small set, a few hundreds, of pixels. For this reason the extraction of 3D coordinates is made by a fetching service, getting the correct coordinates from the depth data stored in the bottom layer.

**Compression** The increase in storage space required for all of these mipmaps is a third of the original texture, because the sum of the areas  $1/4 + 1/16 + 1/64 + 1/256 + \dots$  converges to  $1/3$ . To contain file size, the colour images, the depth images, containing the float array, and the matrix data are archived in a compressed stream.

## 2D Visualization

An ad hoc viewing algorithm has been developed in order to display a multi resolution image. The goal is to draw the multi resolution image in a canvas and allow the user to pan or zoom the image using simple operations.

The first problem in visualization is data loading: the algorithm loads the correct tiles from the multi resolution pyramid during the navigation, starting from some visualization parameters, such as the zoom factor and image position within the canvas. Each level of the pyramid has a different resolution, a different degree of detail. The algorithm chooses the correct level taking into account the canvas resolution, the screen dpi and the tile size. The initial level fits exactly the screen and the image is divided by  $width = height = 2^{maxLev}$ . In this way the initial level  $lev$  is 0 if  $windowWidth < tileWidth$  or  $windowHeight < tileHeight$ . Otherwise the following rules are applied:

$$lev = \min(levX, levY)$$

$$2^{levX} \cdot tileWidth < windowWidth < 2^{levX+1} \cdot tileWidth$$

$$2^{levY} \cdot tileHeight < windowHeight < 2^{levY+1} \cdot tileHeight$$

The next step takes care of the tiles alignment within the canvas. The following pseudo-code snippet introduces the variables  $offsetX$  and  $offsetY$ . These variables indicate the distance between the top left corner of the image and the top left corner of the window.

$$offsetX = \frac{windowWidth - 2^{lev} \cdot tileWidth}{2}$$

$$offsetY = \frac{windowHeight - 2^{lev} \cdot tileHeight}{2}$$

As described before, the initial state includes all tiles of a level *lev*. Once the user interacted with the canvas, the viewer has to move or zoom the image accordingly toward a set direction. The algorithm deals with selecting a new set of tiles, updating the view parameters, that are the value of *lev* and the offsets, so the paint method will have the correct values and the multi-resolution image will be shown properly. Let's introduce the `getTileFromCoords()` method, that returns a tile information given the *x* and the *y* coordinates.

```
getTileFromCoord(int x, int y) {
int col = floor(x/tileWidth);
int row = floor(y/tileHeight);
return getTileInformation(lev, row, col);
}
```

In order to obtain the correct set of visible tiles for the current view, the algorithm selects all and only tiles of the current level with the *row* and *col* components included between the *row* and *col* values of the tile that contains the top left corner of the window, and the *row* and *col* values of the tile that contains the right bottom corner (figure: 12.3(a)).

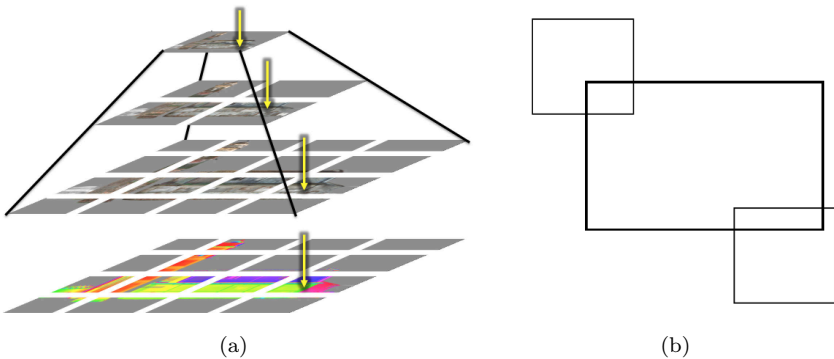


Figure 12.3: a) Accessing the correct tile of the depth image, at bottom level. b) Example of the top-left, and bottom-right tiles.

Before starting the paint method, the algorithm checks whether some tiles are not legal, for instance if the tile is external to the image, then it applies some corrections. Finally the paint method draws all visible tiles, leaving out all the others. The variables *beginCol*, *endCol*, *beginRow* and *endRow* define the set of tiles belonging to the visible region. The alignment algorithm is omitted because it can be easily found in common multi resolution viewers [29].

```
for ( iColumn = beginCol; iColumn <= endCol; iColumn++ ) {
```

```
for ( iRow = beginRow; iRow <= endRow; iRow++ ) {  
    ...  
    print (iRow, iColumn);  
}  
}
```

## Memory management

As explained before, the algorithm uses only necessary tiles and to optimize the loading time. However there is still a big problem: the viewer is designed to work in a web platform and hypothetical delays due to a limited bandwidth, could negatively affect the image exploration because the viewer can't draw a tile that has not yet been loaded. The solution to this problem is simple and widely used: the tile to be loaded is temporarily replaced with the proper portion of one that has already loaded, less resolution, tile. The algorithm keeps the memory clean by disposing of some tiles once they leave the canvas's window. Thanks to this improvement on the cache's management, the viewer ensures a very smooth and fast navigation.

## 3D Coordinates Extraction

The previous section describes the viewer's algorithms supporting the 2D exploration of a multi-resolution solid image. With some interactions, the user can access the 3D coordinates of the point cloud by clicking on the canvas and moving the mouse pointer. When the user clicks on a canvas's point, the viewer shows the 3D coordinates, in the global reference system (world coordinates), of the point under the mouse pointer. To retrieve the 3D world coordinates, the system needs to find the correct tile in the bottom layer of the multi-resolution pyramid containing the 2D mouse coordinates; then the 2D mouse coordinates  $(x, y)$  in the canvas's reference system are transformed into the reference system of this tile  $(dx, dy)$ . This information is needed since depth data are stored only in the bottom layer of the pyramid as to avoid data redundancy. After these few simple steps it's possible to access the depth file corresponding to the correct tile, and then the depth value is retrieved by making a query with the coordinates in the tile's reference system  $(dx, dy)$ . In the last step the tile's coordinates  $(dx, dy)$ , along with the depth value, are converted in world coordinates  $(x, y, z)$  accessing to the matrix stored in the bottom layer. As described before, this matrix is a combination of the view, the projection and the world matrix. Applying simple matrix operations the coordinates of the window are converted into the world coordinates. As shown in figure 12.3(b), finally the system returns the 3D point corresponding to the pixel the user interacted with.



### 12.3.3 User Experience

To allow the exploration of a multi-resolution solid image away from a workstation, a viewer and web service were developed. The proposed solution only requires an internet connection and a common web browser. Once a compressed solid image is created, the web service uploads the corresponding archive onto the server and all the uploaded images are displayed. The user can now choose an image by selecting a list item. Accessing the multi-resolution pyramid, the viewer loads the smallest level that fits entirely on the screen, generally the level 0 or 1, depending on screen resolution and dpi.

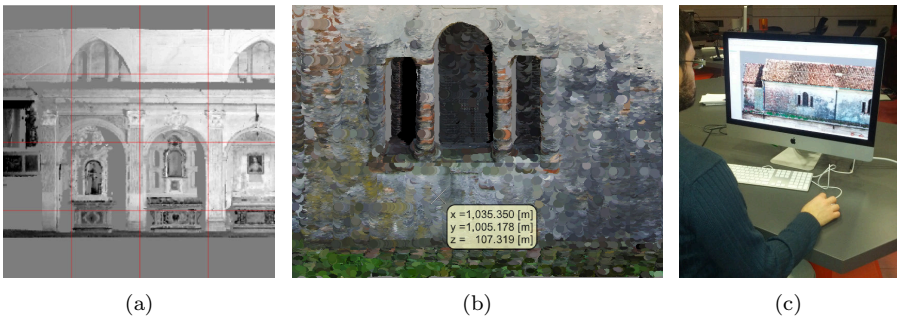


Figure 12.4: a) The image is divided into different tiles. b) The 3D coordinates are displayed in a rounded frame. c) Exploration of a solid image using desktop setup.

**Standard Desktop Platform** When using a desktop platform, the interaction is the very simple: drag and drop for pan, scroll for zoom. The pan operation changes only the offset parameters, while the zoom operation is more complex: in addition to changing the level of detail it updates the offset in order to center the origin of the zoom transformation in the point under the mouse pointer. Therefore this point stays in the same position after the scroll interaction has occurred and the user can easily zoom a detail.

**Multitouch Experience** In order to provide an unencumbered experience and a more natural exploration of the solid image, the web viewer supports multi touch interaction. The navigation is designed like the most popular multi touch interfaces: dragging while pressing one finger allows to pan, and pinch gestures zoom in and out 12.5(a). A gaussian animation ensures a smooth zoom operation providing a smooth and flowing interpolation from one level of detail to another. In this way civil engineers can open a project and explore some images using their personal smart phone or tablet. This feature is highly desirable because lets a technician work on the construction site, away from

his/her home or office. In addition to the description of possible practical cases, the proposed approach also supports a collaborative work. It's possible to show some information to a group of people using a mobile device at any time. Indeed, users are able to engage in a large variety of working activities, while exploring information on the solid image viewer, both individually and in team. Furthermore, the web solution allows for an easier sharing of solid images by simply forwarding the web link of the solid image to other people.

**3d coordinates picking** The viewer obviously gives the user the chance of accessing the 3D coordinates related to the image's pixels. Pressing a finger in the same point for a small amount of time, a flat rounded frame appears in the canvas, reporting the 3D coordinates of that point (12.4(b)). Examining the spatial information in conjunction to the color appearance, the users can better understand the shape and discriminate the model's silhouette resolving all the possible ambiguities. Another advantage is that civil engineers can analyze the 3D coordinates to compare the CAD project to the real scan and monitor the work progress.

**Distances** The UI is designed to provide additional useful information. On pressing a specific button, the viewer can change its layout and enter in the measure mode. When this mode is enabled, the canvas shows two holders connected by a line that represents the distance between the two points specified by those holders. Thanks to this interface, the user can place holders in order to get the spatial distance that exists between two points of the image. During user interaction, the distance is shown in a frame placed in the middle of two holders reporting the length in millimeters (figures 12.5(a) and 12.5(b)), . The distance is calculated only if the first and the second point are legal (don't belong to the background). Pressing the self-explanatory button 'Save distance' the user saves the currently displayed distance, thus having a means to see and correct this measurement at a later time.

### 12.3.4 Discussion

This work is a step forward to a satisfactory exploration of a complex 3D model. To contain the interactive task's load, we are motivated by a simple idea: replacing a slice of a massive point cloud with a simple image while preserving all the 3D information. In addition, our approach exploits a multi-resolution technique with the aim of:

- Reducing the memory usage. The proposed algorithm estimates the displayed portion of the image and then loads only a subset of smaller pictures.

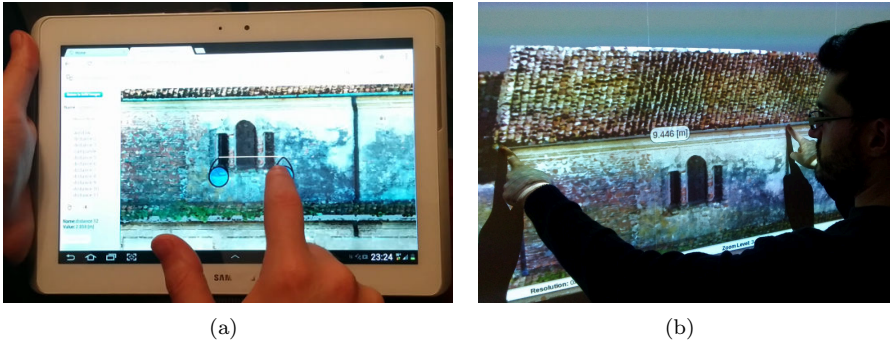


Figure 12.5: By using simple gestures, the user explores the image and takes a measurement in different multi-touch environments: a tablet (a) and an interactive wall (b)

- Improving the quality. Rendering large textures where only small subsets of points are used can easily produce moiré patterns [2].
- Speeding up rendering times, since smaller textures are loaded and unloaded.

Regarding the solution of splitting the original image into smaller tiles, it is encouraging to notice that the creation and navigation steps use an alternative way of merging and displaying the images. They don't load an amount of memory proportional to the total number of the pixels, but an almost constant RAM occupation. This proposal not only reduces the system stress during the visualization process (as explained in the previous section), but it overcomes the limits imposed by the graphics cards. Theoretically we can produce solid images of infinite resolution, the only limits are the creation time and the hard drive free space. In order to contain the file size, all the image's files are compressed and during navigation the viewer decompresses the stream at runtime. Much effort has been devoted to the design of a web solution that allows the exploration of a multi-resolution solid image away from a workstation. Thanks to the multi touch web viewer, users can exploit their ability to interact with their bare hands. If compared to 3D models, it is easier to manipulate an image and the overall interactive impact is hence, reduced. For these reasons our interface does not require any specialized pre-learned skills. An immediate practical consequence is that civil engineers can work on site, away from his/her home or office, using their own mobile device. Indeed, our proposal provides an efficient environment where users can engage in a large variety of working activities while exploring information both individually and collaboratively. Last but not least, the web nature of our solution offers an easier way of sharing models.

Furthermore, the measurement tools developed are particularly useful to engi-

neers that aim to monitor the progress of work: for instance they can analyze the 3D coordinates comparing the CAD project to the real scan.

However, that vision is far from complete. The viewer shows only the colour images and the 3D information is accessed when the user presses and holds his/her finger on the screen. A better solution might be the display of the depth image superimposed over the colour image to better distinguish the model's silhouette and its contours. Further improvements will be aimed at analyzing the depth information of the solid image to extract a valid vectorial 2D curve. Another possible solution could be the rendering in the web viewer a 3D model, reconstructing it from 3D points stored in the solid-image. This can be achieved in two steps: an off-line process refines the solid image depth data outputting an optimized model, then a detailed three-dimensional surface is drawn by using the WebGL library. This way, users would be able to speed up their work interacting with fine grained projects.

**Acknowledgements:** The multi-resolution solid image technique was developed for *Gexcel S.r.l* company in collaboration with M.Pisu Information Technology Engineer. Part of this work are based on his Second Bachelor Degree thesis.

## 12.4 Exploring massive point clouds using multi-touch gestures

As seen before, 3D models are quite difficult to manage for the inexperienced user. In the previous section we examined a way to explore a slice of a 3D model using a 2D image. This method is widely used in Civil Engineering, when you want to record a number of measurements to be attached to a project. This practice is however not good enough/adequate in other applications. To support the user in the exploration and inspection of massive point clouds we developed a multi-touch manipulator.

The hardware consists of a multi-touch table in “closed box” version [23] with modified optical sensor according to the technique reported in section 6.1. The interaction paradigm we used is quite standard within the multi touch interfaces designed for exploring 3D contents. The supported gestures are:

- Rotating: rotation that occurs by placing a finger on the table's sensing surface and moving it in one direction.
- Zooming: dragging two fingers in opposite directions zooms the image
- Panning: placing more than two fingers on the surface and dragging in the same direction moves the viewpoint.

As figure 12.6(a) shows, the application was used during an exhibition to display 3D scans acquired via laser scan technologies. Users showed a lot of

interest (figure 12.6(b)) because the viewer displayed detailed contents, at the same time allowing them to go ahead with the exploration.

A smooth navigation of a point cloud provide a way to create useful material. For instance, technicians could explore the models taking ‘as-built’ images and measures. The final viewer allows for the creation of images and videos, and provides at the same time a method for the creation of distances. These tools are particularly useful, especially in the construction field, for the natural exploration of a point cloud acquired through a laser scanner.

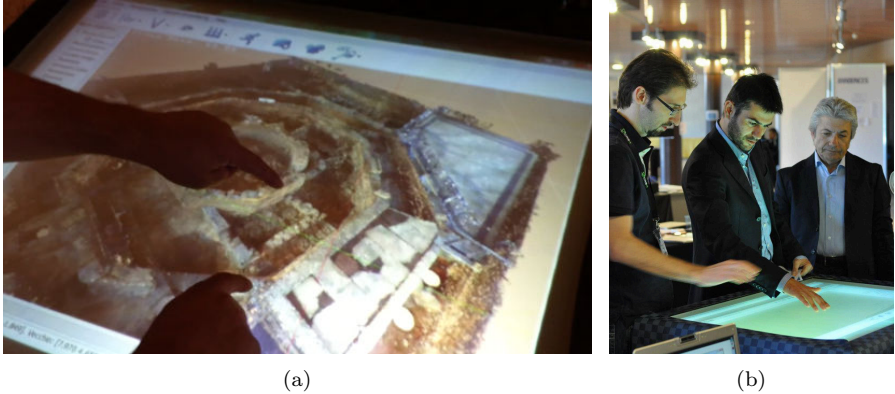


Figure 12.6: a) A user explores a highly detailed 3D point cloud by means of simple gestures. b) Even the mayor of Cagliari was attracted by our interactive installation.

**Fairs and exhibitions:** The closed box multi-touch table was unveiled at Eurographics 2012 from May 13th to May 18th in Cagliari, where the users had the chance to explore the massive 3D models and large point clouds. The application development was based on the graphic engine of *Gexcel R3* software.

## 12.5 Free-hand interaction supporting volume calculation of digital elevation models

In the last few years the increasingly high quality of detailed point clouds generated by recent 3D scanning equipment led researchers to consider a new set of engaging features. One of the most important features is the estimation of the volume described by three-dimensional models, which represent a terrain’s surface. Elaboration on high resolution digital elevation models (DEM) is, however, a very challenging task, since we need to face the limited performance of common graphics platforms. Brute force methods cannot handle such a big 3D scene complexity, hence we need to design more adaptive techniques. This

section introduces an innovative approach based on a tiled-rendering of a multi-resolution point cloud. Each tile is rendered as a range image corresponding to a different portion of the point cloud. Finally the volume is estimated taking into account the distances described by all range images. The proposed solution is not processor intensive and fully exploits current GPU potential. We also worked on designing a competitive free-hand interface to provide a more natural 3D exploration. In case of models representing the bottom of a lake or a valley our interface supports the volume estimation. This way, users can use a double hand gesture to mimic the level of water while the interface reveals the current estimated value of the storage volume.

### 12.5.1 Digital Elevation Models

Real-time 3D exploration of digital elevation models (DEM), which is a digital 3D representation of a terrain's surface [2], is now used in a number of different practical applications, such as landscape modeling, geographic information systems and scientific simulations. Used for the digital production of relief maps, DEMs are commonly built using data collected using sensing techniques, such as photogrammetry and LiDAR, but they can be enhanced from land surveying. The DEM are usually represented by a raster, a grid of points, also known as a 'range image' or a 'heightmap' when representing elevation. This grid of points is also considered as a 'structured point cloud' because made of a set of 3D points stored in a  $N \times M$  matrix and it is possible to go back to the original structure arranging the data according to the original grid. Recent technologies can acquire high quality models, both in terms of accuracy and high resolution. This grid of points is therefore very dense and very close to the original terrain. This is the reasons why DEMs are used in a range of different applications. DEMs may be useful for landscape modeling, city modeling and visualization applications but they are also often required for flood or drainage modeling, land-use studies and geological applications. In landscape modeling the accurate reproduction of the terrain's surface allows the analysis and measurement of some of the original terrain properties without needing to go there to take measurements. By using a computer, for example, users are able to measure the width of roads and bridges to monitor their state. One of the most requested features is the calculation of volume. If a DEM represent a valley, it is possible to the estimate the amount of water contained in the water reservoir. The volume calculation is also useful for earthworks to optimize the cut and fill process. In earthmoving, cut and fill is the process of constructing a railway, road or canal whereby the amount of material from cuts roughly matches the amount of fill needed to make nearby embankments, so minimizing the amount of construction labor. In mining the volume calculation is used to estimate the extracted amount of rock or to check the quantities of soil or rock contained in a cave. Therefore DEM's accuracy and resolution are of primary importance. The higher the DEM's resolution, the more accurate

the extracted measurements and estimated volume. The DEMs high resolution, however, leads to a number of problems: firstly it is hard to render a high resolution 3D model, supporting an interactive and reactive exploration at the same time using common graphics platforms. Secondly, the analysis of a huge data set, aiming to estimating the volume for example, is not trivial both in terms of processing time and of used memory.

## 12.5.2 Visualization of large-scale high resolution terrain

Visualization of large 3D models has been an active topic since graphic systems were invented. During all this time, developers and researchers had at their disposal platforms with limited performances. Much of their work successfully addresses culling and sub-sampling techniques to render a realistic virtual scene using commodity graphics platforms. Since vision is generally considered the most dominant sense, the development of an effective rendering system that preserves the high quality of the models is of primary importance. Therefore, sub-sampling methods could not provide a satisfactory level of detail. More efficient methods for generating high quality visual representations of virtual environments are required.

Several solutions adopt a service-oriented system architectures, based on high performance, server-side 3D viewer which are interactively visualized by corresponding thin clients, such as smart phones or common desktop computers. Although the WVS proposals [81] decouple the clients from the complexity of 3D city models, they have, however, large disadvantages: the rendering of massive 3D models requires specialized and expensive hardware in the server platforms; since they are server-based, their service is available to a very limited number of clients; moreover, 3D clients based on perspective views can only provide a limited interactive 3D experience to the user.

To overcome all these problems, we used a different technique that sets up more distributed systems. Innovative works [28, 47–49] propose an efficient approach for construction of multi resolution structures supporting interactive visualization of very large point clouds. Thanks to a simple efficient recursive clustering method, an off-line process computes a hierarchy of point clouds. Each cloud in the hierarchy is a pre-computed simplified version of the original model. At rendering time the clouds are combined coarse-to-fine to locally adapt sample densities according to the distance between the point of view and the projected model. The resulting system allows rendering of complex models at high frame rates using consumer graphics platforms. Moreover, various techniques improve the rendering quality. To draw a continuous surface that better fits the real terrain, for example, the 3D points are replaced with splats oriented towards the normal vector. Several improvements in the cloud construction step allows the merging of different clouds in one model, aligning them with each other. If the model has only a gray scale color, the intensity, fast methods implement the colouring using external colour images: the intrinsic and extrinsic parameters

of the images are found after a calibration step; then, in a off-line process, a blending algorithm projects a rectified image into the point cloud, colouring the three-dimensional points. The dynamic nature of the rendering algorithm made possible the realization of a client-server application. Fetching the data from the network, the client's viewer shows simple 3D points. During navigation, it transmits some information about the current view to the server, such as the field of view or the frustum, then the server returns the 3D coordinates of the points that fall in that frustum, accessing the hierarchy of the multi resolution structure. Different tests show how the application remains usable even for very large models on consumer-level network connections.

### 12.5.3 Volume calculation

Beside the efficiency offered at rendering time by the multi resolution structure, the development of fast algorithms that work on the entire point cloud is not easy task. The volume calculation is an explicative example. Given an altitude value, we want to calculate the volume included between the given altitude and the point cloud that describes the terrain. Since DEMs ensure that each point of the cloud represents the altitude of the terrain in this particular position in the earth, standard approaches run through all points, subtracting, for each point, the value of the given altitude from the value of the cloud's altitude. The sum of all these differences is then multiplied with the 2D area covered by each point. Since DEMs are regular grids, the calculation of the area is very simple. Assuming that a DEM has grid of  $1500 \times 1000$  points and that it covers an area of 3 by 2 kilometers, the area covered by each point has a width of  $3000/1500$  meters and a height of  $2000/1000$  meters. Therefore, a single point covers an area of 2 by 2 meters. Finally the area is multiplied by the sum of all the differences of altitudes previously obtained and the correct value of the volume is calculated. This method is however, CPU intensive and cannot handle a high resolution point cloud. Besides the time spent by this brute force algorithm to examine the entire model, the DEM can be so large that it can not be loaded in a array stored in RAM.

We developed a more efficient method based on a tiled-rendering of a multi resolution point cloud. The main idea is to take a series of shots, dividing the original grid of points in smaller rectangles and creating smaller heightmaps, or tiles. Each heightmap is then processed independently from each other, providing a correct estimation of the total volume. Assuming that we have already constructed a multi-resolution point cloud according to the algorithm described in the previous section, the first step consists in the calculation of the heightmap size. According to the graphics card specifications, such as the maximum size of the frame buffer, and taking into account the point density of the DEM, the algorithm chooses the size of each tile. The correct point of view from which to take each shot is then calculated. Since we can keep active only



one OpenGL context in the graphic card, all tiles are rendered sequentially. In the last step the heightmaps are processed in parallel: for each tile the algorithm calculates the volume analyzing all its points and the total volume is calculated as the sum of each tile volume.

Theoretically the proposed approach allows the calculation of the volume processing a DEM of infinite resolution. The point cloud multi resolution structure ensures an off-core fetching of data, therefore only a small portion of the model is loaded onto memory at runtime. Moreover the tiled strategy processes simultaneously different smaller portions of the DEM grid, reducing the computation time.

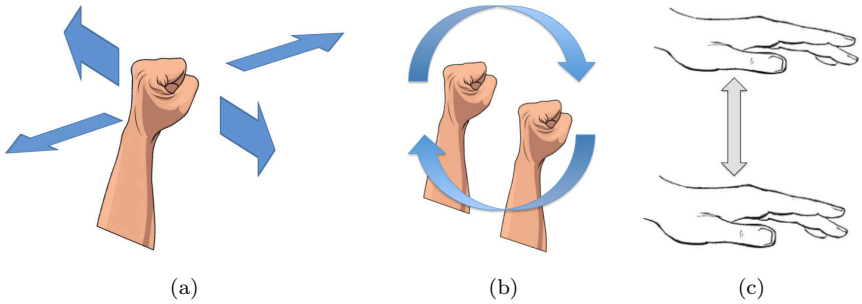


Figure 12.7: a) Panning in the 3D space by means of a single hand movement. b) Performing on-air dual handed gesture, the user can rotate the 3D scene. c) The volume calculation is supported by moving away the hands.

### 12.5.4 Free-Hand Interaction

Thanks to the efficiency in terms of computation time offered by both the rendering approach and the DEMs volume calculation, we were able to develop a free-hand interface supporting real time interaction with digital elevation models and volume calculation. Using a Microsoft Kinect sensor, our interface provides a free-hand interaction with the virtual scene, exploiting grab gestures that have to be performed by the user. Therefore, the user starts the interaction with the virtual terrain simply mimicking the act of grasping a real object.

#### 3D exploration of relief maps

The user can interact with the terrain visualization rotating the scene around its barycenter and moving it towards or away from him/her in order to get the desired level of detail.

The proposed interaction paradigm is very simple:

- pan is supported by on-air grab action with either the right or left hand. Keeping the hand closed and simultaneously changing its position, the user can move the model up and down, back and forth, left and right (figure 12.7(a)).
- rotation is supported by performing on-air dual handed gesture with both hands closed. By making a circular movement with both hands, the user can rotate the scene around its barycenter (figure 12.7(b)).

After defining the interaction paradigm, we tried to set up a comfortable interactive space. A large desktop display shows a high-quality rendering of a dense terrain map. The width provided by the screen gave us the opportunity to build a functional interaction scenario where the user can navigate in the 3D space by means of simple gestures, exploring the virtual terrain and even focusing on a detail.

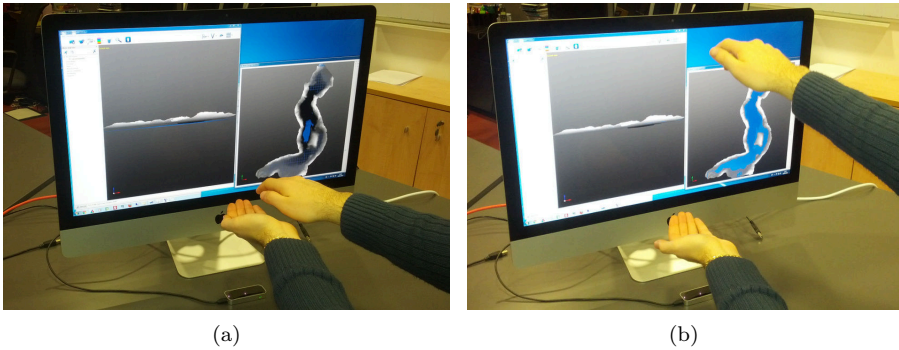


Figure 12.8: Distancing their hands, users can bring the water level from low (a) to high (b).

## Volume Calculation

The interface we designed for the volume calculation is slightly different. A standard interface includes a scrollbar that is used to input and change the value of water level. We propose an alternative interaction paradigm, based on free-hand gestures. Our intention is to help the user by speeding up the input of the water level value using both hands.

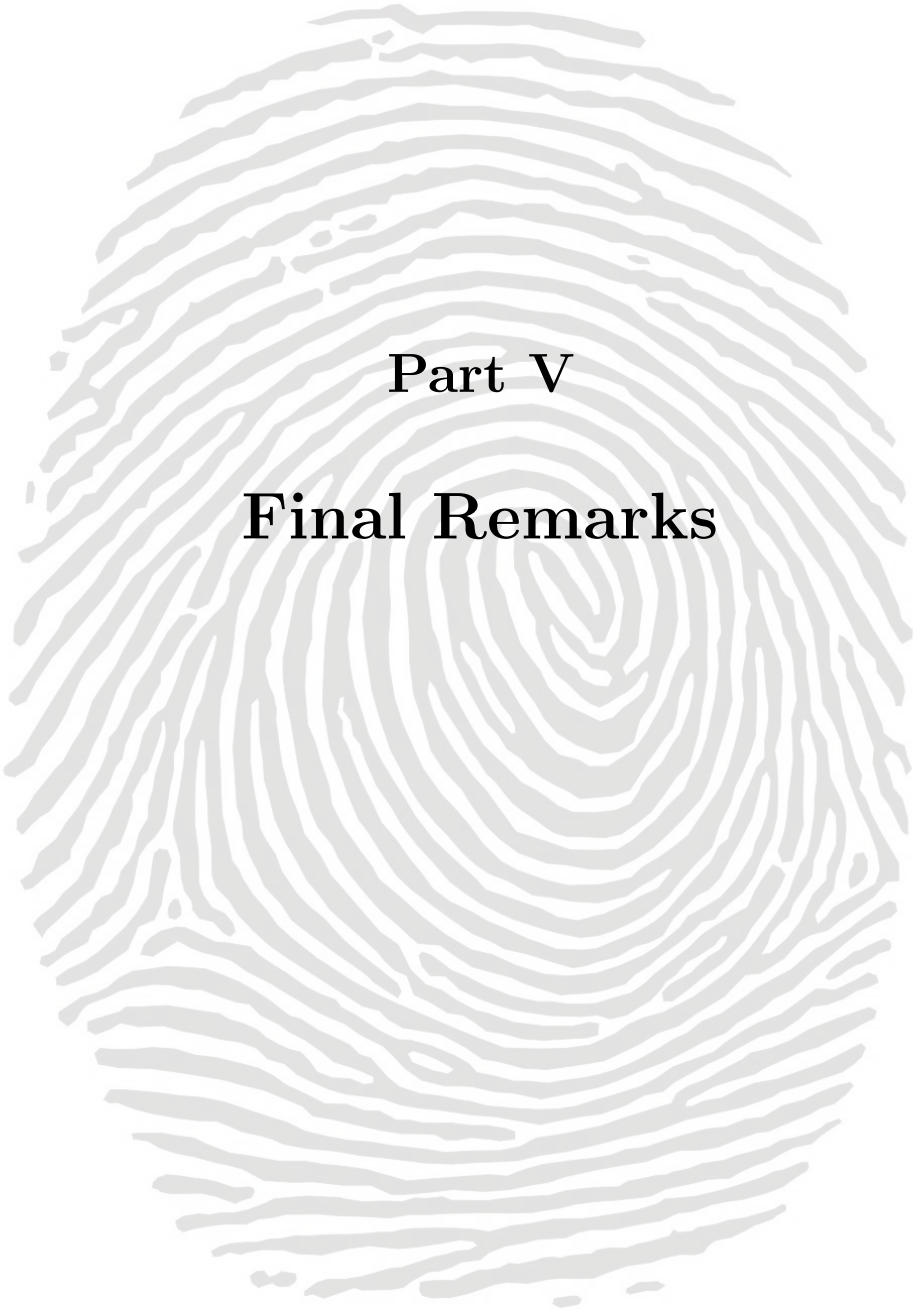
For this reason we developed an interface that senses the user position and reacts when a hand movement occurs, as seen in figures 12.8(a) and 12.8(b). The user has to keep one hand steadily open to represent the lake bottom, while his/her other hands stands for the water level. To increase this level the user needs to raise his hand; on the contrary if the hand is lowered the value decreases.

Exploiting the human co-ordination abilities and its measurement perception, the input of the levels' values is highly improved. A hand facing down in horizontal position represents the level of the water, whereas up and down hand movement represent a change of the water level.

## 12.6 Summarising

In this chapter we investigated the 3D exploration of large-scale high resolution 3D models. Our study is threefold. First of all we analyzed the visualization of a dense point cloud. Since the user's desire is to interact with attractive 3D models and that, especially in engineering applications, the 3D model must be significant and must show all its essential details, we adopted a multi-resolution approach to render the point cloud. Secondly, we designed a multi-touch interface to explore these high resolution models. Then, we tried to improve the algorithms for 3D models volume estimation. According to a tiled rendering, the described approach allows the real time computation of volumes, even when using large point clouds. Finally we proposed an alternative interface that exploits free-hand gesture to support the exploration of large map reliefs and the volumes calculation. Users are able to use both their hands to grab the model with the intent of moving and rotating it, similarly to the way people do in the real world. Furthermore, our interface supports dual hand gestures for the volume calculation of water storage. The user needs to open his/her hand, placing it horizontally, to mimic the level of the water. This way, people can exploit their own coordination skills and interact with the application.





Part V

Final Remarks



# Chapter 13

## Conclusions

The current evolution trend in the Human-Computer Interaction field endeavours to create instruments that can be readily and naturally usable. What makes an interaction technique “natural” is still being debated in the scientific world: the use of inborn and pre-acquired skills and the collaborative aspects are of primary importance. From this point of view, the investigation on the human gesture communication and real objects’ manipulation capability is doubtless very useful. To begin with, we tried to build efficient prototypes to do experiments and deploy alternative interfaces.

Firstly, in order to examine the effectiveness of manipulative action, we began with the development of a tangible support, as described in Chapter 5. This experiment required building and augmenting an interactive table where some objects were laid. By picking an object the users can get some information about it: in fact when a certain object is lifted, the system sets in motion a video associated with the object itself. Although the interaction is very simple, almost trivial, this first experiment helped us to get our first impressions on the importance of manipulative actions. As various tests with real users suggested, the possibility of controlling the application through manipulation of real objects, familiar to the users, not only facilitates the interaction, but also encourages and attracts people to use the application. This appreciation is mainly due to two aspects: users handle real objects performing an action of picking, commonly used in the real world.

Then, in Chapter 6 we analyzed multi-touch devices and interfaces with the intent of producing a gestural and manipulative system. Although various low cost approaches obtained the approval of the international research community, we deemed necessary the adjustment of technologies so that prototypes can be positioned in uncontrolled lighting environments, like open spaces, museums,

fairs and exhibitions, places fit for hosting a collaborative type of work.

People use their senses to gather and interpret the physical world, and the tools exploiting those abilities are the most effective. This is mainly the reason why we investigated on free-hand and full-body interaction. Since man is used to move around in a real 3D space, we developed a free-hand gestural interface, based on the tracking of body parts movements, to explore models and 3D scenes. As explained in Chapter 7, the important considerations made on tangible interaction influenced the way we designed this interface. In fact, we exploited the act of opening and closing the hand, which resembles the act of grasping (picking) a real object, to map the selection task of 3D interaction. That is, the user has to close at least one hand and then move it in the real 3D space to zoom, pan, or move a object in the 3D scene.

Our investigation on manipulations and gestures continues in Chapter 9. In the real word there are several scenarios where gestures and manipulative action prevail. Let's think about the browsing of hardcopy documents, about the ways we search for a photograph or a document. We usually spread the material on a table and shuffle around with our hand to visually search until we find the photo or the document we were looking for. On the contrary in a common desktop setup, we are forced to explore visual contents using mouse and keyboard that limit our manipulation ability. To overcome this limit, in the first part of this chapter, we reported on an alternative interface that helps accessing of visual documents through a combination of free-hand gestures and manipulations. This way, the designed interface exploits a way of browsing through digital documents that is closer to common people skills. Another aspect of primary importance involving gestures, is communication. Gesticulation helps to better express the concepts we are trying to communicate, supporting and strengthening the information that is being expressed. This is why gestures play a crucial role in teaching, because they encourage audience attention and learning. For this reason the rest of Chapter 9 is focused on a system that can help assessing the educational performances, employing a full-body gestural interface to analyze the teachers' movements during a presentation or a lesson in the classroom.

A man-oriented interactive space not only allows the removal of man-computer barriers, but also facilitates and stimulates people's communication. We proved this by building an interactive social notice board (chapter 10) where people could use the relevant interfaces to keep in touch and accomplish collaborative tasks like organizing and classifying a set of photographs, videos, and notes collected during a group holiday. What we learnt in the previous parts of this dissertation is finally used here, starting from the adoption of reliable multi-touch sensor devices, the definition of a co-operative environment, up to the exploitation and comparison of manipulative and gestural actions.



**Evaluation.** In chapter 11, with the aim of evaluating the ‘naturalness’ and the efficiency of gestural interfaces, but also comparing different interaction paradigms, we reviewed specific tasks. After conducting small-scale users tests we discussed the observations on user behaviour.

Firstly, we observed the task of pair-programming, performed at a traditional desktop versus a multi-touch table. Thanks to this experiment we verified that, while working at the multitouch table, people perform the task of understanding and debugging algorithms better than at a traditional desktop. This is due to the fact that when using a multi-touch setting there is an increased amount of non-verbal communication (gestures, body postures, facial expressions, etc.) than when adopting a traditional desktop.

Then, we described a different experiment with the intent of comparing multi-touch and free-hand interfaces. We chose to compare the two interfaces analyzing the 3D exploration of a virtual planetarium, continuing what we started in Chapter 7. This required the building of two different interaction scenarios. The first is based on a simple multi-touch table, while the second one exploits a free-hand interaction together with a projection on a geodetic sphere. From the results we concluded that people prefer the multi-touch environment for its more fine-tuned control offered by direct manipulation. Taking into account other factors, such as the increase of sensory and realism perception for the user, the full-body system is definitely better.

**Case study.** In the end, we were ready to exploit the lessons learnt so to design gestural interfaces aimed at specific application fields. Since we had already gained a lot of experience in designing interfaces devised for exploration of 3D models and scenes, we decided to take under review applications in the Architecture and Construction field. In this context the objective was to find innovative solutions that would allow the natural exploration of high resolution tridimensional models, that represent entire buildings, squares and terrain surfaces, to ease engineers and technicians work. Since vision is generally considered the most dominant sense, we firstly addressed our efforts on the development of a rendering system that preserves the models high quality. We then developed applications that, via the use of multi-touch and free-hand gestural interfaces, stimulate the exploration of large virtual scenes, allow the inspection of extremely detailed 3D models and provide tools for measuring distances and volumes.

With the intent of applying a more natural approach to other application fields, in Appendix A we introduced tiny observations on interactive music.

**Future Work.** To begin with, we plan to work on increasing both the performance of the developed sensors and the applications interaction capabilities. From the experiments carried out about 3D exploration (chapters 7 and 11) it emerged that , when adopting free-hand gestural interfaces, people feel that the application as a bit inaccurate in terms of precision. Obviously, the lack of precision of the sensing device does not help. But this is mainly because a tactile feedback is not available, contrary to what happens in multi-touch or tangible interfaces where the direct-manipulation plays a role of primary importance.

To overcome this limitation, we intend to exploit more devices for controlling the interfaces (e.g. the Leap Motion for a more precise hand tracking). Alternatively it is also possible to modify the interfaces. Just to give an example, a visual feedback like a pointer on the screen showing our hand position can be used to overcome the lack of a tactile feedback.

Future work will be aimed at designing an interactive environment even more suitable for co-operative and collaborative tasks. This can be accomplished, for example, applying and evaluating what we reported in Chapter 8. Recognizing users and tracking their position in order to move interface items according to user movements without needing to touch them. Implementing such features by means of inexpensive depth camera, a user could have his/her own personal workspace always close at hand.

Moreover we have to study a technique that allows the transfer of contents from user to user, maybe just using simple gestures.

In future work we will also explore new applications for such technologies that suit both the way people interact with objects and the way people interact with each other. The social and collaborative dimensions will be explored in a scenario of interactive leisure/learning, such as an interactive museum. This will bring advancement on the recognition of more abstract gestural languages [71] to control the system, in order to support richer and more exciting interaction.

## 13.1 Contributions

The core contribution of this thesis is in gestural interaction for which various technical solutions and interfaces have been developed. For each topic, we reported the challenges faced in building these environments, the solutions we came up with and the lessons learnt.

In particular, this work gives a strong contribution to the state of the art in the multi-touch and free-hand interaction field. The development of interactive walls, multi-touch tables and full-body interaction scenarios is necessary for comparing and evaluating the different interaction techniques.

We chose low cost approach and commodity hardware to cut down on costs, adapting the technologies, where possible, with the aim of improving sensing

performances.

Here is a list of prototypes we contributed to create.

**Closed box multi-touch table.** This prototype developed by some of our colleagues [23] was completed with a high contrast projecting surface and more adherent to the acrylic glass. In the final version we replaced the LED frame with an infrared illuminator placed on the bottom of the box, going from FTIR<sup>1</sup> approach to a DI<sup>2</sup>, applying the improvements described in the 6.1 section to the new DI setup.

**Open box multi-touch table.** Our approach, described in detail in section 6.1, allowed us to build a proper table setup because the closed box is no longer needed. This prototype uses two projectors and the resulting display shows very large and sharp images. The proposed technique enabled us to position the table in different locations without having to redesign the lighting system while the prototype simplicity, a base and a top, has helped with transport, assembling and disassembling operations.

**Interactive Wall.** In the final stages of this doctorate we had the privilege of working on a 3 mt wide interactive wall developed by the NIT's CRS4 team<sup>3</sup>, improving the technology according to what we reported in section titled 6.2.

**Full-body interactive stellarium.** With the aim of enhancing the virtual planetarium experience and mimicking the visualization of the sky ceiling, we developed a full-body interface by building a hemispherical surface where the sky map is projected onto (section 11.2).

### 13.1.1 Software frameworks

During the drafting of this dissertation, the source code of various applications has been re-written right from the start, with the intent of developing cross platforms and well-written frameworks, testing different programming strategies, like the policy-based design and design patterns. Most of the material was published on the Internet as open-source, sharing the challenges we faced and the solutions found with other people. This made our work intended not only for technicians and researchers, but also for the wider community of amateurs in the hope that additional revisions can fine-tune the solutions we found so far.

---

<sup>1</sup>Frustrated Total Internal Reflection [55]

<sup>2</sup>Diffuse Illumination

<sup>3</sup>Natural Interaction Technologies at the Center for advances studies, Research and development in Sardinia

## Interactive Viewer

With the passage of time we designed and fine-tuned a framework that allows the development of application for the browsing of city maps, notes, photos and images, videos and 3D models.

Aiming to support alternative surfaces, we developed free-hand and multi-touch manipulators. We carefully selected quick rendering techniques such as FBOs and shaders, but also space partitioning data structures, like kd-trees and quad-trees. We also included some network functionalities [35, 75] to support communication with sensors, on line streaming and access to individual contents through the most famous social networks APIs.

We can see some of the framework applications in these sections: 8, 9.1, 10.

## Multi-touch sensor

Considering the effective limitations of existing multi-touch frameworks and libraries [touchlib, OpenTouch, Touchkit] we developed a sensor software for multi-touch tables that include the changes described in the section 6.1. Our framework supports multiple optical technologies based on infrared light, the recognition of visual markers<sup>4</sup> and tangible objects put on the surface as described in section 8. We used this sensor in every multi-touch application discussed in this dissertation.

Summarizing, the work carried out can provide some useful indications about the way we addressed gestural interaction. All these prototypes enabled us to test and compare the developed interfaces and the different interaction paradigm, carrying out whenever it was possible, small-scale user tests (sections 11.1, 11.2), and then analyzing users' performance.

## 13.2 Fairs and Exhibits

We point out how extremely important is, in the interaction field, the testing with real users. All interfaces and prototypes developed were widely tested by a number of people in labs.

In some cases we relocated the prototypes, from the room where they were designed to fairs, installations and exhibitions where we conducted small-scale test with inexperienced and casual users.

---

<sup>4</sup>QR-code and fiducials

- We took the closed box multi-touch table to Eurographics 2012 from May 13th to May 18th in Cagliari, where we analyzed the users' performance in the exploration of massive 3D models and large point clouds, as described in section 12.4.
- A transportable version of the *PickARock* tangible experiment was installed at the "Sardinien for alle sanser" for 8 days in Copenhagen in October 2013 (chapter 5).
- Devised for a permanent exhibition, the fixed installation of *PickARock* was hosted within the photographic exhibition for the promotion of the Sardinian Store in Berlin in December 2013 for 20 days (chapter 5).
- The open box multi-touch table and the interactive wall were shown during various seminars and student open days in CRS4 labs <sup>5</sup>.
- Regarding the evaluation of gestural communication in teaching, we needed to acquire a consistent dataset of people movements, going around to visit schools and filming teachers' performances (section 9.2).
- In order to compare the users' performance in pair programming (chapter 11.1) and 3D interaction tasks (chapter 7), we took our multi-touch table and free hand interaction scenario to the atrium and public spaces of *Palazzo delle Scienze*<sup>6</sup> of the University of Cagliari.

During these events, many people used the appliances. We noticed that the users asked to try out the applications, participating voluntarily to tests and questionnaires and giving suggestions related to their own area of interest, particularly useful for the further improvements of proposed solutions.

This participation is partly due to the fact that many of these technologies were perceived as novelty. On the other hand, the demos involved the visitors through the use of multiple senses facilitating a larger cooperation.

**Tangible Interaction:** As far as tangible interfaces are concerned, users widely appreciated the possibility of controlling the appliance through manipulation of the real objects, and by using multiple senses they felt more involved.

**Multi-touch Interaction:** During the demonstration with the interactive wall, various people, especially the newcomers, were surprised by the fact that they had to interact with the normal wall, with simple white plaster. What's more, the large dimension of the multi-touch table allowed for an important co-operation among themselves.

---

<sup>5</sup>Open Media Center Lab - Building 1- Loc.Pixinamanna, 09030 Pula (Ca), Italy

<sup>6</sup>Department of Mathematics and Computer Science, University of Cagliari, via Ospedale, 72 09124 - Cagliari, Italy

**Free hand Interaction:** People enjoyed and appreciated manipulating the 3D models by mimicking with their hands the real movements of grasping. Free-hand manipulation allow the inexperienced users to inspect 3D objects at various scales, integrating panning, rotating, and zooming controls into natural and simple operations.

### 13.3 Published as

Parts of this dissertation have been published before.

S. A. Iacolina, A. Soro, R. Scateni. *Improving FTIR Based Multi-touch Sensors with IR Shadow Tracking*. Proc. Of EICS 2011 ACM SIGCHI Symposium on Engineering Interactive Computing Systems. Pisa Italy 13-16 June 2011. [67]

E. Tuveri, S. A. Iacolina, F. Sorrentino, D. Spano, R. Scateni. *Controlling a planetarium software with a Kinect or in a multi-touch table: a comparison*. Proceedings of CHIItaly '13: ACM SIGCHI Italian Chapter International Conference on Computer-Human Interaction , p.6, September 2013, Trento, Italy. [124]

S. A. Iacolina, M. Corrias, O. Pontis, A. Soro F.Sorrentino, R. Scateni *A Multi-touch Notice Board Fostering Social Interaction*. Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI, p.13, September 2013, Trento, Italy. [65]

S. A. Iacolina, A. Soro and R. Scateni. *Natural exploration of 3D models*. In Proc. Of the 9th Conference of the ACM SIGCHI Italian Chapter. Sept 13-16, 2011. Alghero (Italy). ACM, New York, NY. [85]

A. Soro, S. A. Iacolina, R. Scateni, S. Uras. *Evaluation of User Gestures in Multi-touch Interaction: a Case Study in Pair-programming*, 13th International Conference on Multimodal Interaction - ICMI 2011. [119]

M. Careddu, L. Carrus, A. Soro, S. A. Iacolina, and R. Scateni. *Moravia: A video-annotation system supporting gesture recognition*. SIGCHI - CHIItaly 2011. Alghero (Italy). Adjunct Proceedings. [90]

**Other papers** have been published as part of the activities involved in this work, but have not been included in the final discussion.

S. A. Iacolina, A. Lai, A. Soro, R. Scateni. *Natural Interaction and Computer Graphics Applications*, EuroGraphics Italian Chapter 2010, pp. 141-146. Genova, Italy, November 2010. [66]

V. Vacca, M. N. Iacolina, A. Pellizzoni, S. A. Iacolina, A. Trois, and the SRT Astrophysical Validation Team. *CASTIA - A Source Visibility Tool for the Italian Radio Telescopes*, Internal Report INAF - IRA 468/13, <http://www.ira.inaf.it/Observing/castia/site/>, May 2013. [126]





## Chapter 14

# Personal Notes

Music has always been a great companion in my life. Ever since I was a child I've been immersed in it, playing piano for as far as I can remember, I can't recall a day in my memories where music would not flow through my mind or my hands. Over the years I have never stopped playing, and actually, fate has given me the gift of being able to express my emotions through music, pouring anger, happiness, sadness on the piano keys. My thoughts could be heard through the vibrations of the instrument's chords, as if they were words.

I think that like music and musical instruments, computers should also be a medium of communication, helping us expressing ourselves and facilitating the conveyance of our feelings to others.

Drawing our conclusions we turned our research to the interactive spaces because we do not have at the state of the art a complete system that best supports any task. We tried to develop environments that offer a balanced compromise between collaborative and manipulative reality.

We developed easy to use interfaces or *tools*, targeted to a particular mix of applicative fields, avoiding generalized interfaces or *workbenches* that make people's actions difficult.

With the objective of knocking down the barriers between man and computer we developed environments that also facilitate man-to-man communication, purposely designed for team work, where the natural interfaces are distributed in a way that favors and encourages collaboration.

Having considered the tests carried out with novices and casual users we can

say that the natural interfaces should be distributed into space, transforming them from alternative interactive mechanisms to definitive bespoke interfaces aimed to the resolution of specific and contained problems.

From this perspective the interactive environments are seen as a reference point for people of any age and social class, as to encourage physical activity, communication and mind training. In other words they become an aggregation place where they can pursue personal and collective objectives.

# Appendices



# Appendix A

## “Seeing” the Music

A wee bit of this doctorate has been devoted to musical interaction. We investigated two themes in particular. We tried in the first place to make the participation to a live concert more captivating creating visual effects linked to the melody during the musicians’ performance. Secondly, we implemented a multi touch interface to control the playback of a musical track.

Unfortunately there has not been an extensive enough research for an entire chapter on this subject, but we hope to re-discuss the topic in the near future.

### A.1 A “musical” lamp

We analyze in this section what we experience during a gig or a live performance. Let’s think about performances held by famous musicians and groups in arenas or other venues and the way we feel when attending these events. The performers on stage are in the limelight and sound engineers are continuously adjusting the audio, so that we can enjoy the best sound quality.

Not only that but more skilled technicians are in charge of lighting, adjusting the direction, colour, intensity, flooding the stage and the audience. Attending these events is usually very exciting. However during live performances with my band called *Ritmofficina*, we realized how such an experience could be dramatically enhanced either for the attending audience or for the musicians. Yes, it is true, as we have just said, in live concerts we obviously get both music and visual effects but the latter are usually not linked to the music that’s being played. From a melody point of view they are random, lights change their colour without being associated to the sound being played.

To overcome this problem we created a floor lamp that analyzes the tonality of an audio signal and emits lights associated with that tonality. The prototype

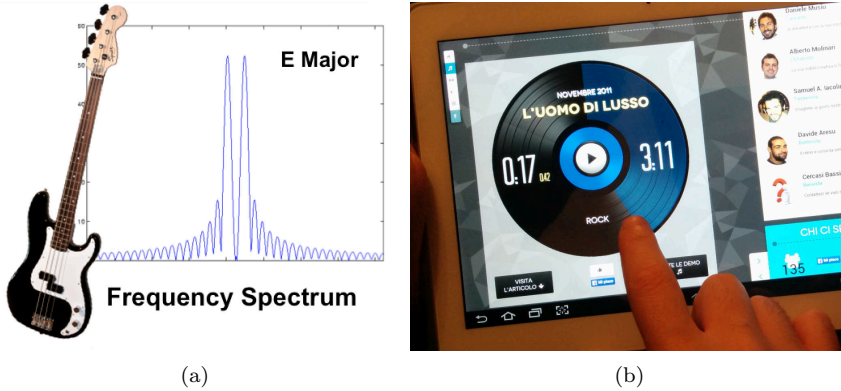


Figure A.1: a) A simple algorithm filter the audio signal coming from the bass, determining the raw frequency and the played note.

was developed connecting a bass to the line input of a sound card. A simple algorithm filters the audio signal detecting the note that is being played by the bass player (figure A.1(a)). The computer then, once it analyzes the tonality, changes the colour of an RGB illuminator positioned in the lamp shade. The last step consists of establishing a protocol, associating each note of a chromatic musical scale to a different colour.

Let's describe now, how this prototype behaves. In figure A.2(a), the bass player does an E major and the lamp goes green. When the tonality changes to an A major, the lamp changes to blue (figure A.2(b)). Since this system only scans a single audio signals, the keyboard player (the author) can play along without interfering on the lamp functioning.

This system implies a number of innovations. In indoors exhibitions, in clubs and pubs, where there isn't enough staff and budget to afford a stage-like light system, the musicians themselves can control visual effects by means of a simple lamp.

We can point out how a musical tonality is linked to a colour tone, and additionally, for the ones with some familiarity with music, this effect is extremely pleasant.

What's more, even someone who is hard of hearing can be even more intensely involved. They can in fact feel the vibrations coming from the instruments and from the loudspeakers and “see” the music: colours reflect the music, not only as rhythm but also as melody.



Figure A.2: a) Players execute E major and the lamp lights up in green. When they move to a A major tonality the lamp change its color, showing a bright blue.

## A.2 Multi-touch music player

We will now describe a multi touch interface created for controlling the playback of a musical track. Through simple actions the users can “seek” in a track by acting on the graphic vinyl. Every touch takes the start time to exactly that position whereas moving their hand can go back and forth (figure A.1(b)). The interface is compatible with any web browser, either in mobile phones, tablets and desktop computers.





## Appendix B

# Human Computer Interaction

We can't talk about interaction and interfaces without briefly describing the perceptive, sensory and cognitive abilities of the human being. As we will see, the study of these abilities is the main domain of interaction activities between man and machine. This first dissertation is going to be the basis for following chapters and also for the development of an interactive environment.

### B.1 Main objectives

The object of investigation of *Human Computer Interaction* is often re-defined. By analyzing what happened in the past, we can surely state that history is influenced either by the knowledge of man's cognitive capabilities and by the machine's technological capabilities. Based on this, researchers consider HCI as applied social science <sup>1</sup>, that can facilitate people's activities and lives. In the last few years, the study of interaction analysed in depth the human cognitive system and socio-cultural factors, moving the field of application from computer to society.

#### B.1.1 Brief History

For years the main objective of HCI psychology was the understanding of man's cognitive abilities subject to interaction. This cognitive aspect shows what the new way of working will be in the long run: traditional ergonomics

---

<sup>1</sup>S. Bagnara and S. Pozzi, "Fundamentals, History and Trends of HCI" (S.A. Iacolina translation of "Fondamenti, Storia e Tendenze dell'HCI", pg. 17 [118])

addressing the man's physical strain, will disappear thanks to the fact that tasks demanding physical and mental effort will be performed by machines and highly automated systems; the physical labour is now done by the machine only. The cognitive effort comes along when the use of an interface is required to control the machinery: the man's job is now decision making and interaction with the interface that controls the machinery. This new approach to work implies a change of habits on the workers side. It is therefore the evolution of computers and their pervasive presence in the work place that marks the real birth of HCI.

Since 1970 we've seen the rising of the so called knowledge society, characterized by an ever growing homogeneity between working and living environment and by an extensive use of information technologies and automation.

In 1990 Grudin [54] compared HCI to computer's evolution pinpointing five fundamental moments. Herewith is the list that highlights the different definitions of man-computer interface over the time.

In the first level we have large dimension devices, whose interface was made of just the internal circuits and various switches that allowed the programming of the device. This kind of interface can be considered just of hardware type: the users of this device were engineers (if not the same inventors of the machine) who had the hardware know-how.

The second level illustrates what happened during the first half of the 70's. The first CRT monitors were far too expensive still for widespread use. As a consequence the software was represented by a list of printed codes or by perforated cards. The users of such interface were IT programmers, and the fundamental requirement for using these systems was the knowledge of disciplines of strictly information technologies type.

The third level is the appearance of the very first interface: a terminal. Its purpose is to display on a screen inputs entered by the user and translate these inputs into commands for the computer circuits. This system applies the very first abstraction from the machine physical details. The users of this kind of devices don't need having technical-scientific exclusive knowledge to be able to use the machines: the definition of end user is created to indicate that the device was suitable for anyone with basic information technology expertise.

The fourth level is the launch of the personal computer (PC): graphic interfaces were introduced in the common computer user's life. These types of interfaces of higher level allow for a more natural and man friendly dialogue with the computer.

The last level described by Grudin is the transition from the study of interaction between computer and individual, to the study of the computer as a support to work groups (Computer Supported Cooperative Work - CSCW). The Computer increasingly becomes an instrument used primarily to communicate and the HCI in just a few decades takes its field of application from the calculator to the social environment.

We gather, from this scenario, that the knowledge of how the machine actually works is not important for the interaction, since the interface translates the user's actions into inputs for the computer's circuits. We therefore move from overspecialised users like electronic engineers or programmers, to common users that use computers at work but also in their leisure activities.

## B.1.2 Recent Challenges

HCI today's challenges are definitely represented by the attempt to address issues within the social context, that add up to the other challenges previously described. There are different social dynamics that HCI need to face, and they have something in common: there are tensions between various situations resulting from technologic innovation.

### From Need to Desire

The first issue we will analyse is the link between need and desire. The word 'work' is the fulcrum of classic ergonomics and the origin of HCI: the scope is the reduction, removal where possible, of fatigue in working environments, not only physical but also the disease that can result from the interaction with machinery.

However, recently HCI is applied also to leisure and entertainment fields [7]. HCI is used to satisfy the needs of users and deliver pleasurable experiences. One of these is the evolution of mobile phones: today's level of innovation achieved by portable devices like tablets and smartphones wouldn't be explicable as a simple improvement of existing communication portable devices. The new technologies support our needs, our desires: increased portability, larger memory, holding an ever growing quantity of higher quality information and multimedia applications.

The HCI cannot just assess the interaction in terms of efficiency and performance, it also needs to satisfy the users' subjective aspects that relate to usage experience like the aesthetics of the product for instance, but also the emotional involvement, sense of satisfaction, amusement resulting from its use and last but not least the easiness of use. These are the aspects that the user experience (UX) study focused on.

Technologies are still quite hard to use and they often generate frustration rather than satisfaction. HCI must contribute in a decisive way in order to simplify the spread of digital literacy. This must be done in a short period of time because being work, leisure time and social life all aspects that share a common infrastructure called technology, whoever won't be able to master this infrastructure will be left out not only from the workplace but also from social and community life.

Work also evolves into new forms from complex production processes, organized mainly in the same way, based on the repetition of operations and on production

lines, to a varied dynamic process, flexible, intellectually and socially engaging. On the day to day basis there is a growing trend to multitask, the working day is longer and intense and very often we run on both lines, never releasing the tension between work and leisure time. The HCI still has to give a satisfactory answer to the negative aspects in this change.

### **Information appliances**

Anything can be digitalized. Any kind of information, video, audio, text, or images, can be codified. Digitalisation allows the integration of different information in the same device, increasing the complexity and decreasing the degree of specialization.

New generation computers but also a number of digital personal devices fall into this classification including smartphones, iPods, netbooks, tablets, eBook readers; all of them personal and individual appliances, but as complex as a computer with functions that tend to overlap.

On the other hand, a recent branch of HCI intends to find out the techniques that will spread intelligence in the environment, typically by the use of appliances meant to carry out specific tasks (information appliances). Its main task is having to face and study the two opposing trends, between an increasingly personal computer and various specialized devices.

### **Information Overproduction and Overconsumption**

The increase of data transmission speed and data storage capability reduces the cost of producing, transmitting and storing digital information. This creates a new trend that needs to be addressed: information overproduction and over consumption. People usually store more information than what they can indeed use, therefore creating a conflict between information storage and viewing: in a few hours you can acquire more information than a scientist in 1700 could get during his entire life.

The HCI must create new interfaces to control the fruition of information, providing new instruments for grouping, researching and visualizing information.

### **Browsing**

This in return created new, faster search devices that allow us to not only skip the browsing our folders but also avoid the creation of proper organized archives altogether. Such instruments prevent us from the coming across relevant information by chance, since we only find whatever we already know we need to find. In other words we face additional antagonism, this time round between information search and browsing.

## The human attention

The consequence of over-storing information and the impossibility of consuming all of it, is also that data become quickly outdated and obsolete and is no longer worthwhile investing on it: let's think about the bad feeling left by web pages that were clearly last updated years ago.

Attention is a limited resource and data constantly fights to get it, with new inputs quickly replacing the old ones. Just think about the way we read the News: the media applications in smartphones (available for main newspapers) show us endless lists of headlines with short abstracts from the news, forcing us to quick browsing of the headlines and eventually, only if really interested, read the full article.

## From calculation to communication

Over the years we've also witnessed the evolution in the use of computers: from mere calculation tasks to communication. Technologies are more and more a useful vehicle of social interactions, rather than having the original function of powerful calculation. The opportunity of being constantly connected to the internet generates security problems. The use of social networks discloses our our preferences and habits, allowing the tracing of a string of digital information that leads to the vulnerability of concepts like privacy and anonymity.

## B.2 Interfaces

The objective of this paragraph is the mentioning of models that had major importance in the history of HCI with reference to design and study of interfaces, focusing our attention on the prototypes evaluation step.

### B.2.1 Interaction

The HCI is based on a very careful observation and study of human nature and on the fact that perception and action are not two separate states, focusing our attention on the instruments' role in mediating the human action. Perception is involved in the selection of actions, in their execution and also in the continuous valuation of results. In other words an action cannot be considered complete without evaluation stage.

## B.3 Designing Interfaces

The designing of interactive interfaces requires a great effort in the development process of IT instruments<sup>2</sup>. Designing an interface that can be used by different users of different age and experience is not easy. The interface must therefore be submitted to various opinions, potentially contrasting. It is mainly for this reason that planning of interactive interfaces is an experimental and pioneering activity for which development methodologies, evaluation techniques were specifically devised [39] and new approaches like contextual design [6] are continuously introduced.

The principles to be adopted for an ergonomic design, focused on the end user, are various and have been formulated in different ways. However, to date, the four pioneering principles expressed by Gould e Lewis [53] still represent a valid reference system to whom, any newer interpretations refer to.

Design, implementation and evaluation were traditionally considered as separate phases of the man-machine systems' development process. One of the major contributions in designing interactive systems was the introduction of the reiterative project development concept, where project and assessment are repeated until a satisfactory result is achieved. The evaluation stage completely encompasses the process: it is necessary to evaluate the existing system when planning, but also the human activity and the context where this activity is carried out, the prototypes created and the final system.

## B.4 Evaluation

The evaluation is one of the essential steps of project designing focused on the end user. In this phase, through usability tests, case studies, questionnaires and other methodologies you can reiterate the assessment cycle in order to introduce adjustments to the system until you achieve the end user requirements. In the project design and development through sequential prototypes, tests on the prototype are carried out at every stage of development. Tests include two different types of checks:

- *verification*: check on the product consistency with specifications requirements;
- *validation*: check that the product is meets the purpose it was conceived for.

The co-validation activities makes sure that the final prototype complies with the (stated or sometimes still unstated) user's or client's expectations. Hence,

---

<sup>2</sup>R.Polillo, "Introduction to the Engineering Usability" (S.A. Iacolina translation of "Introduzione all'Ingegneria dell'Usabilità"), pg. 115 [118]

the co-validation can't be done by designers only (as it often happens during verification activities), but it also requires the user's involvement. That's why they are more difficult and critical than the verification activities. As they say, it is the difference between making the thing right (verification) and making the right thing (co-validation).

In order to perform/carry out these validation assessments we may use different techniques, the most used of which are grouped under two different categories:

- Assessments carried out by experts in usability, without user's involvement. These evaluations can be named *inspections*. The most popular are the so called *heuristic evaluations*.
- Evaluations carried out with the user's involvement, including *usability tests*, the most important and commonly used.





# References

- [1] E Agosto, I Picco, and F Rinaudo. THE SOLID IMAGE WEB VIEWER: A NEW WAY FOR 3D SURVEY DATA WEB-FRUITION.
- [2] Philippe Beaudoin and Pierre Poulin. Compressed Multisampling for Efficient Hardware Edge Antialiasing. In *Proceedings of the 2004 Graphics Interface Conference, GI '04*, pages 169–176, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [3] Michel Beaudouin-Lafon. Lessons learned from the WILD room, a multi-surface interactive environment. In *23rd French Speaking Conference on Human-Computer Interaction, IHM '11*, pages 18:1—18:8. ACM, 2011.
- [4] Hrvoje Benko and Andrew D Wilson. DepthTouch: Using Depth-Sensing Camera to Enable Freehand Interactions On and Above the Interactive Surface. Technical Report MSR-TR-2009-23, Microsoft, 2009.
- [5] Hrvoje Benko, Andrew D Wilson, and Ravin Balakrishnan. Sphere: multi-touch interactions on a spherical display. In *Proceedings of the 21st annual ACM symposium on User interface software and technology, UIST '08*, pages 77–86. ACM, 2008.
- [6] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-centered Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
- [7] Susanne Bødker. When second wave HCI meets third wave challenges. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, pages 1–8. ACM, 2006.
- [8] Richard A Bolt. "Put-that-there": Voice and gesture at the graphics interface. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, New York, NY, USA, 1980. ACM.
- [9] Leandro Bornaz and Sergio Dequal. THE SOLID IMAGE: a new concept and its applications. XXXIV:78–82.

- [10] D Bowman, E Kruijff, J LaViola, and I Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley., Boston, 2005.
- [11] Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics, I3D '97*, pages 35—ff., New York, NY, USA, 1997. ACM.
- [12] Doug A Bowman, Ernst Kruijff, Joseph J Laviola, and Ivan Poupyrev. An Introduction to 3-D User Interface Design. In *Presence: Teleoperators and Virtual Environments*, pages 96–108, 2001.
- [13] Christopher J Bradley. *The Algebra of Geometry: Cartesian, Areal and Projective Co-ordinates*. Highperception, 2007.
- [14] J Brooke. SUS: A "quick and dirty" usability scale. In P Jordan, B Thomas, and B Weerdmeester, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis, London, 1996.
- [15] Barry Brown and Matthew Chalmers. Tourism and mobile technology. In Kari Kuutti, EijaHelena Karsten, Geraldine Fitzpatrick, Paul Dourish, and Kjeld Schmidt, editors, *ECSCW 2003 SE - 18*, pages 335–354. Springer Netherlands, 2003.
- [16] Duane C Brown. Decentering Distortion of Lenses. *Photogrammetric Engineering*, 32(3):444–462, 1966.
- [17] Pascal Bruegger and Béat Hirsbrunner. Kinetic User Interface: Interaction through Motion for Pervasive Computing Systems. In *UAHCI '09: Proceedings of the 5th International on Conference Universal Access in Human-Computer Interaction. Part II*, pages 297–306, Berlin, Heidelberg, 2009. Springer-Verlag.
- [18] Dimitrios Buhalis and Rob Law. Progress in information technology and tourism management: 20 years on and 10 years after the Internet—The state of eTourism research. *Tourism Management*, 29(4):609–623, August 2008.
- [19] Dimitrios Buhalis and Maria Cristina Licata. The future eTourism intermediaries. *Tourism Management*, 23(3):207–220, 2002.
- [20] Nicholas Burtnyk, Azam Khan, George Fitzmaurice, Ravin Balakrishnan, and Gordon Kurtenbach. StyleCam: interactive stylized 3D navigation using integrated spatial & temporal controls. In *Proceedings of the 15th annual ACM symposium on User interface software and technology, UIST '02*, pages 101–110, New York, NY, USA, 2002. ACM.
- [21] Bill Buxton. Multi-Touch Systems that I Have Known and Loved (Overview). 2007.

- [22] William Buxton and Brad A Myers. A study in two-handed input. *SIGCHI Bull.*, 17(4):321–326, April 1986.
- [23] Daniela Cabiddu, Giorgio Marcias, Alessandro Soro, and Riccardo Scateni. Multi-touch and Tangible Interface: Two Different Interaction Modes in the Same System. CHItaly 2011 Adjunct Proceedings, 2011.
- [24] J M Carroll. *Human-Computer Interaction in the New Millennium*. ACM Press, New York, 2002.
- [25] Chiarello E. Casasola M, Cohen LB. Six-month-old infants’ categorization of containment spatial relations. In *Child Development*. 1987.
- [26] Ginevra Castellano, Loic Kessous, and George Caridakis. Emotion Recognition through Multiple Modalities : Face , Body Gesture , Speech. pages 92–103.
- [27] I Chaudhri. Animated graphical user interface for a display screen or portion thereof, 2010.
- [28] P Cignoni, F Ganovelli, and E Gobbetti. Interactive out-of-core visualisation of very large landscapes on commodity graphics platform. *Virtual Storytelling. . . .*, 2003.
- [29] Sylvain Contassot-vivier, E N S Lyon, and I N P Grenoble. Multiresolution approach for image processing. pages 1–9.
- [30] Susan Wagner Cook and Susan Goldin-Meadow. The Role of Gesture in Learning: Do Children Use Their Hands to Change Their Minds? *Journal of cognition and development*, 7(2):211–232, 2006.
- [31] Susan Wagner Cook, Zachary Mitchell, and Susan Goldin-Meadow. Gesturing makes learning last. *Cognition*, 106(2):1047–1058, 2008.
- [32] P. Knoth D. Herrmannova. Visual Search for Supporting Content Exploration in Large Document Collections. *Volume 18, Number 7/8*, 2012.
- [33] Kelly L Dempski and Brandon Harvey. Supporting Collaborative Touch Interaction with High Resolution Wall Displays. In *Proceedings of the 2nd Workshop on Multi-User and Ubiquitous User Interfaces (MU3I)*, 2005.
- [34] Michael B Denlinger and Haworth NJ. Ambient-light-responsive touch screen data input method and system, 1988.
- [35] Massimo Deriu, Gavino Paddeu, Alessandro Soro, and G Paddeu M Deriu A. Soro. XPlaces: An Open Framework to Support the Digital Living at Home. In *Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*, GREENCOM-CPSCOM ’10, pages 484–487, Washington, DC, USA, 2010. CRS4 – (accepted at IEEE/ACM IOTS 2010), IEEE Computer Society.

- [36] Paul Dietz and Darren Leigh. DiamondTouch: A Multi-User Touch Technology. 3(2):219–226.
- [37] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. The MIT Press, new editio edition, 2004.
- [38] Richard O Duda and Peter E Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM*, 15(1):11–15, 1972.
- [39] Joseph S Dumas and Janice C Redish. *A Practical Guide to Usability Testing*. Intellect Books, Exeter, UK, UK, 1st edition, 1999.
- [40] Andreas Dünser and Eva Hornecker. An Observational Study of Children Interacting with an Augmented Story Book. In Kin-chuen Hui, Zhigeng Pan, RonaldChi-kit Chung, CharlieC.L. Wang, Xiaogang Jin, Stefan Göbel, and EricC.-L. Li, editors, *Technologies for E-Learning and Digital Entertainment SE - 31*, volume 4469 of *Lecture Notes in Computer Science*, pages 305–315. Springer Berlin Heidelberg, 2007.
- [41] C R Dyer. Parallel Computer Vision. chapter Multiscale, pages 171–213. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- [42] Florian Echtler, Manuel Huber, and Gudrun Klinker. Shadow tracking on multi-touch tables. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 388–391, New York, NY, USA, 2008. ACM.
- [43] George W Fitzmaurice, Hiroshi Ishii, and William A S Buxton. Bricks: laying the foundations for graspable user interfaces. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 442–449, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [44] Rita Francese, Ignazio Passero, and Genoveffa Tortora. Wiimote and Kinect: gestural user interfaces add a natural third dimension to HCI. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12*, pages 116–123. ACM, 2012.
- [45] D Fryberger and R G Johnson. Touch actuable data input panel assembly, 1972.
- [46] Ron George and Joshua Blake. Objects, Containers, Gestures, and Manipulations: Universal Foundational Metaphors of Natural User Interfaces. 2010.
- [47] Enrico Gobbetti and Fabio Marton. Layered point clouds. *... of the First Eurographics conference on Point- ...*, 2004.

- [48] Enrico Gobbetti and Fabio Marton. Layered point clouds: a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models. *Computers & Graphics*, (July 2004):1–28, 2004.
- [49] Enrico Gobbetti and Fabio Marton. Far voxels: a multiresolution framework for interactive rendering of huge complex 3D models on commodity graphics platforms. *ACM Transactions on Graphics (TOG)*, pages 878–885, 2005.
- [50] S Goldin-Meadow, H Nusbaum, S D Kelly, S Wagner, and Nusbaum H Kelly S D & Wagner S Goldin-Meadow S. Explaining math: Gesturing lightens the load. *Psychological Science*, 12(12):516–522, 2001.
- [51] Wagner S. Goldin-Meadow S, Nusbaum H, Kelly SD. Explaining math: gesturing lightens the load. *Psychological Science*, 12:516–522, 2001.
- [52] A. N Gopnik, A., Meltzoff. The development of categorization in the second year and its relation to other cognitive and linguistic developments. In *Child Development*, pages 1523–1531. 1987.
- [53] John D Gould and Clayton Lewis. Designing for Usability: Key Principles and What Designers Think. *Commun. ACM*, 28(3):300–311, 1985.
- [54] Jonathan Grudin. The computer reaches out: the historical continuity of interface design. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 261–268, New York, NY, USA, 1990. ACM.
- [55] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pages 115–118, New York, NY, USA, 2005. ACM.
- [56] Jefferson Y Han. Multi-touch interaction wall. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Emerging technologies*, page 25, New York, NY, USA, 2006. ACM.
- [57] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, Bill Buxton, and Michel Pahud Nicole Coddington Jenny Rodenhouse Andy Wilson Hrvoje Benko Ken Hinckley Koji Yatani. Pen + touch = new tools. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pages 27–36, New York, NY, USA, 2010. ACM.
- [58] Hal Hodson. Leap Motion hacks show potential of new gesture tech. *New Scientist*, 218(2911):21, 2013.
- [59] Michael S. Horn, R. Jordan Crouser, and Marina U. Bers. Tangible interaction and learning: the case for a hybrid approach. *Personal and Ubiquitous Computing*, 16(4):379–389, 2012.

- [60] Eva Hornecker and Jacob Buur. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 437–446, New York, NY, USA, 2006. ACM.
- [61] Bradford Hosack. Video{ANT}: Extending Online Video Annotation beyond Content Delivery. *TechTrends*, 54(3):45–49, 2010.
- [62] Juan Pablo Hourcade, Benjamin B Bederson, Allison Druin, and François Guimbretière. Differences in Pointing Task Performance Between Preschool Children and Adults Using Mice. *ACM Trans. Comput.-Hum. Interact.*, 11(4):357–386, 2004.
- [63] Stephen W Hughes. Stellarium—a valuable resource for teaching astronomy in the classroom and beyond. *Science Education News (SEN)*, 57(2):83–86, 2008.
- [64] Sungjae Hwang, Myungwook Ahn, and Kwangyun Wohn. Magnetic Marionette: Magnetically Driven Elastic Controller on Mobile Device. In *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion*, IUI '13 Companion, pages 75–76, New York, NY, USA, 2013. ACM.
- [65] Samuel A Iacolina, Michele Corrias, Omar Pontis, Alessandro Soro, Fabio Sorrentino, and Riccardo Scateni. A Multi-touch Notice Board Fostering Social Interaction. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI*, CHIItaly '13, pages 13:1—13:4, New York, NY, USA, 2013. ACM.
- [66] Samuel A Iacolina, Alessandro Lai, Alessandro Soro, and Riccardo Scateni. Natural Interaction and Computer Graphics Applications. In Enrico Puppo, Andrea Brogni, and Leila De Floriani, editors, *EuroGraphics Italian Chapter 2010*, pages 141–146, Genova, Italy, 2010. Eurographics Association.
- [67] Samuel A Iacolina, Alessandro Soro, and Riccardo Scateni. Improving FTIR based multi-touch sensors with IR shadow tracking. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, EICS '11, pages 241–246, New York, NY, USA, 2011. ACM.
- [68] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, New York, NY, USA, 1997. ACM.
- [69] Hirotaka Iuchi, Sakashi Maeda, and Naoyuki Tsuruta. Gesture Recognition using {Self-Organizing Maps and Hidden Markov Model}. *IPSJ SIG Notes, Computer Vision and Image Media*, 2001(36):127–134, 2001.

- [70] Robert J K Jacob, Audrey Girouard, Leanne M Hirshfield, Michael S Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum. Reality-based interaction: a framework for post-WIMP interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 201–210, New York, NY, USA, 2008. ACM.
- [71] R.~H. Jacoby, M Ferneau, and J Humphries. Gestural interaction in a virtual environment. In S.~S.~Fisher, J.~O.~Merritt, & M.~T.~Bolas, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2177 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 355–364, April 1994.
- [72] Chris Plauche Johnson and Peter A Blasco. Infant Growth and Development. *Pediatrics in Review*, 18(7):224–242, 1997.
- [73] P.A. Johnson, C.P., & Blasco. Infant growth and development. *Pediatrics in Review*, pages 224–242, 1997.
- [74] Martin Kaltenbrunner and Ross Bencina. reactIVision: a computer-vision framework for table-based tangible interaction. In *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 69–74, New York, NY, USA, 2007. ACM.
- [75] Martin Kaltenbrunner, Till Bovermann, Ross Bencina, and Enrico Costanza. TUIO: A protocol for table-top tangible user interfaces. In *Proc. of the The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation*, 2005.
- [76] Leonard R Kasday and Plainsboro NJ. Touch position sensitive surface, 1984.
- [77] A Kendon. *Gesture: Visible Action as Utterance*, 2004.
- [78] Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. HoverCam: interactive 3D navigation for proximal object inspection. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*, I3D '05, pages 73–80, New York, NY, USA, 2005. ACM.
- [79] Michael Kipp. Multimedia Annotation, Querying and Analysis in ANVIL. In M Maybury, editor, *Multimedia Information Extraction*, chapter 19. IEEE Computer Society Press, 2010.
- [80] David Kirsh and Paul Maglio. On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science*, 18(4):513–549, October 1994.
- [81] J Klimke and Jürgen Döllner. Service-Oriented Visualization of Virtual 3D City Models. *directionsmag.com*.

- [82] Teuvo Kohonen. The self-organizing map. In *Proceedings of the IEEE*, volume 78, pages 1464–1479, 1990.
- [83] Myron W Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. VIDEOPLACE—an artificial reality. *SIGCHI Bull.*, 16(4):35–40, 1985.
- [84] Gordon Kurtenbach and Eric A Hulteen. Gestures in Human-Computer Communications. In Brenda Laurel, editor, *The Art of Human Computer Interface Design*, pages 309–317. Addison-Wesley, 1990.
- [85] Samuel A Iacolina, Alessandro Soro, and Riccardo Scateni. Natural Exploration of 3D Models. In *Proceedings of the 9th ACM SIGCHI Italian Chapter International Conference on Computer-Human Interaction: Facing Complexity*, CHIItaly, pages 118–121, New York, NY, USA, 2011. ACM.
- [86] Larry Larsen. Interview to Bill Buxton on NUIs. [\url{http://channel9.msdn.com/Blogs/LarryLarsen/CES-2010-NUI-with-Bill-Buxton}](http://channel9.msdn.com/Blogs/LarryLarsen/CES-2010-NUI-with-Bill-Buxton), 2010.
- [87] S K Lee, William Buxton, and K C Smith. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull.*, 16(4):21–25, 1985.
- [88] Sangyoon Lee, Jinseok Seo, Gerard Jounghyun Kim, and Chan-mo Park. Evaluation of pointing techniques for ray casting selection in virtual environments. In *In Third International Conference on Virtual Reality and Its Application in Industry*, pages 38–44, 2003.
- [89] Tim Love. ANSI C for Programmers on UNIX Systems. [\url{ftp://svr-www.eng.cam.ac.uk/misc/love\\_C.ps.Z}](ftp://svr-www.eng.cam.ac.uk/misc/love_C.ps.Z), 2010.
- [90] R. Scateni M. Careddu, L. Carrus, A. Soro, S. A. Iacolina. Moravia: A video-annotation system supporting gesture recognition. In *Adjunct Proceedings of the 9th ACM SIGCHI Italian Chapter International Conference on Computer-Human Interaction: Facing Complexity*, CHIItaly, 2011.
- [91] Paul P Maglio and David Kirsh. Epistemic Action Increases With Skill. In *In Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*, pages 391–396. Erlbaum, 1996.
- [92] J B Mallos. Touch position sensitive surface, 1982.
- [93] Takefumi Matsunaga and Oshita Masaki. Recognition of Walking Motion Using Support Vector Machine. In *Proceedings of ISIC 2007*, pages 337–342, 2007.
- [94] David Mcneill. *Hand and Mind: What Gestures Reveal about Thought*. University Of Chicago Press, 1992.



- [95] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE transactions on systems, man and cybernetics, Part C, Applications and reviews*, 37(3):311–324, 2007.
- [96] Jon Moeller and Andruoid Kerne. ZeroTouch: An Optical Multi-touch and Free-air Interaction Architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2165–2174, New York, NY, USA, 2012. ACM.
- [97] Cecily Morrison, Matthew Jones, Alan Blackwell, and Alain Vuylsteke. Electronic patient record use during ward rounds: a qualitative study of interaction between medical staff. *Critical Care*, 12(6), 2008.
- [98] Leap Motion. Leap. URL: <https://www.leapmotion.com/>[last accessed 2013-02-04], 2012.
- [99] R Mueller. Direct television drawing and image manipulating system, 1974.
- [100] Sundar Murugappan, Vinayak, Niklas Elmqvist, and Karthik Ramani. Extended multitouch: recovering touch posture and differentiating users using a depth camera. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, UIST '12, pages 487–496. ACM, 2012.
- [101] Ahmed K Noor. Potential of virtual worlds for remote space exploration. *Advances in Engineering Software*, 41(4):666–673, 2010.
- [102] Masaki Oshita and Takefumi Matsunaga. Automatic learning of gesture recognition model using {SOM} and {SVM}. In *6th International Symposium on Visual Computing 2010 (Lecture Notes in Computer Science 6453)*, pages 751–760, 2010.
- [103] Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. When it gets more difficult, use both hands: exploring bimanual curve manipulation. In *GI '05: Proceedings of Graphics Interface 2005*, pages 17–24, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.
- [104] James Patten and Hiroshi Ishii. A comparison of spatial organization strategies in graphical and tangible user interfaces. In *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*, pages 41–50, New York, NY, USA, 2000. ACM.
- [105] Vladimir I Pavlovic, Rajeev Sharma, and Thomas S Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):677–695, 1997.

- [106] J. Piaget. *Genetic epistemology*. 1970.
- [107] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 79–80, New York, NY, USA, 1996. ACM.
- [108] PrimeSense. NITE Middleware.
- [109] L Rabiner and B Juang. An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(1):4–16, April 2003.
- [110] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. pages 267–296, 1990.
- [111] Carlos Ricolfe-Viala and Antonio-Jose Sanchez-Salmeron. Lens distortion models evaluation. *Appl. Opt.*, 49(30):5914–5928, 2010.
- [112] D Rowan. Kinect for Xbox 360: The inside story of Microsoft's secret 'project natal'. *Wired Magazine*, 2010.
- [113] K Sabir, C Stolte, B Tabor, and S I O'Donoghue. The Molecular Control Toolkit: Controlling 3D molecular graphics via gesture and voice. In *Biological Data Visualization (BioVis), 2013 IEEE Symposium on*, pages 49–56, 2013.
- [114] R Sagawa, M Takatsuji, T Echigo, and Y Yagi. Calibration of lens distortion by structured-light scanning. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 832–837, 2005.
- [115] Johannes Schöning, Peter Brandl, Florian Daiber, Florian Echtler, Otmar Hilliges, Jonathan Hook, Markus Löchtfeld, Nima Motamedi, Laurence Muller, Patrick Olivier, Tim Roth, and Ulrich von Zadow. Multi-Touch Surfaces: A Technical Guide. Technical Report TUM-I0833, University of Münster, 2008.
- [116] Johannes Schöning, Jonathan Hook, Tom Bartindale, Dominik Schmidt, Patrick Oliver, Florian Echtler, Nima Motamedi, Peter Brandl, and Ulrich Zadow. Building Interactive Multi-touch Surfaces. In Christian Müller-Tomfelde, editor, *Tabletops - Horizontal Interactive Displays SE - 2*, Human-Computer Interaction Series, pages 27–49. Springer London, 2010.
- [117] M Soga, K Matsui, K Takaseki, and K Tokoi. Interactive Learning Environment for Astronomy with Finger Pointing and Augmented Reality. In *Advanced Learning Technologies, 2008. ICALT '08. Eighth IEEE International Conference on*, pages 542–543, 2008.

- [118] Alessandro Soro, editor. *Human computer interaction. Fondamenti e prospettive*. Polimetrica Int. Scientific Publisher, 2008.
- [119] Alessandro Soro, Samuel Aldo Iacolina, Riccardo Scateni, and Selene Uras. Evaluation of User Gestures in Multi-touch Interaction: A Case Study in Pair-programming. In *Proceedings of the 13th International Conference on Multimodal Interfaces, ICMI '11*, pages 161–168, New York, NY, USA, 2011. ACM.
- [120] Alessandro Soro, Gavino Paddeu, and Mirko Lobina. *Multitouch Sensing for Collaborative Interactive Walls*, volume 272, pages 207–212. Springer Boston, 2008.
- [121] Opencv Dev Team. Open Source Computer Vision Library - Reference Manual. [\url{http://opencv.itseez.com/}](http://opencv.itseez.com/), 2011.
- [122] Tammy D Tolar, Amy R Lederberg, Sonali Gokhale, and Michael Tomasello. The Development of the Ability to Recognize the Meaning of Iconic Signs. *Journal of Deaf Studies and Deaf Education*, 13(2):225–240, 2008.
- [123] Yukitaka Toyokura and Yoshihiko Nankaku Et al. Approach to Japanese Sign Language Word Recognition using Basic Motion {HMM}. In *Proceedings of the Society Conference of IEICE*, volume 2006, page 72, 2006.
- [124] Elena Tuveri, Samuel A Iacolina, Fabio Sorrentino, L Davide Spano, and Riccardo Scateni. Controlling a Planetarium Software with a Kinect or in a Multi-touch Table: A Comparison. In *Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI, CHIItaly '13*, pages 6:1—6:4, New York, NY, USA, 2013. ACM.
- [125] Selene Uras, Daniele Ardu, Gavino Paddeu, and Massimo Deriu. Do not judge an interactive book by its cover: a field research. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, pages 17–20. ACM, 2012.
- [126] V. Vacca, M. N. Iacolina, A. Pellizzoni, S. A. Iacolina, and A. Trois. CASTIA - A Source Visibility Tool for the Italian Radio Telescopes. Technical report, Internal Report INAF - IRA 468/13, 2013.
- [127] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. Analysis of the accuracy and robustness of the leap motion controller. *Sensors (Basel, Switzerland)*, 13(5):6380, 2013.
- [128] Pierre Wellner. The DigitalDesk calculator: tangible manipulation on a desk top display. In *UIST '91: Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 27–33, New York, NY, USA, 1991. ACM.

- [129] R M White. Tactile sensor employing a light conducting element and a resiliently deformable sheet, 1987.
- [130] Laurie Williams and Robert Kessler. *Pair Programming Illuminated*. Addison-Wesley, New York, 2003.
- [131] Karl D D Willis, Takaaki Shiratori, and Moshe Mahler. HideOut: Mobile Projector Interaction with Tangible Objects and Surfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, pages 331–338, New York, NY, USA, 2013. ACM.
- [132] Andrew D Wilson. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, New York, NY, USA, 2004. ACM.
- [133] Andrew D Wilson and Hrvoje Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, UIST '10, pages 273–282, New York, NY, USA, 2010. ACM.
- [134] B Witmer and M Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 3(7):225–240, 1998.
- [135] Allison Woodruff, Andrew Faulring, Ruth Rosenholtz, Julie Morrision, and Peter Pirolli. Using Thumbnails to Search the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 198–205, New York, NY, USA, 2001. ACM.
- [136] J.J Woodward, A.L., & Guajardo. Infants' understanding of the point gesture as an object- directed action. In *Cognitive Development*. 2002.
- [137] Ying Wu and Thomas S Huang. Vision-Based Gesture Recognition: A Review. In *GW '99: Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pages 103–115, London, UK, 1999. Springer-Verlag.
- [138] Robert Zeleznik, Andrew Bragdon, Ferdi Adeputra, and Hsu-Sheng Ko. Hands-On Math: A Page-Based Multi-Touch and Pen Desktop for Technical Work and Problem Solving. In *UIST2010*, 2010.
- [139] R Zijlstra and L van Doorn. The Construction of a Scale to Measure Subjective Effort. Technical report, Delft University of Technology, Department of Philosophy and Social Sciences, Delft, Netherlands, 1985.





# Acknowledgements

I would like to thank Prof. Riccardo Scateni for his precious guidance during all the course of this work.

I am sincerely grateful to Marco Fiocco, Alessandro Soro and Marco Campanella, whose constant guidance, encouragement and support, throughout the development of this work, enabled me to produce this PhD thesis. I always considered them as my personal mentors, being of great inspiration to me and I would like to thank them for their precious comments and advice.

My acknowledgement also goes to Dr. Michael Nebeling and Dr. Fabio Paternò, whose invaluable remarks have been of great help to the improvement of this presentation.

Thanks to my family, as without their support this thesis would not have been possible. I must also thank Alberto Molinari for his support and his musical performances.

Last but not least all the people at the Computer Science Lab (aka *Batcave*) and all my colleagues and friends at Cagliari University that made my work easier and pleasant during all these years.

*Cagliari, 2014 April 28*