Università degli Studi di Padova

Department of Information Engineering

Bachelor Degree in Information Engineering

Mixed Reality applications in medical therapy

Supervisor
Prof. Simone Milani
Università di Padova

Co-supervisors
Hannah Luxenberg Tono
Alberto Tono
Fabio Capraro
Sebastiano Verde

Candidate
Elena Camuffo

A.A. 2018-2019

23 Settembre 2019

Alla mia famiglia.
Al nonno Dino,
Al mio Daniele.

"Chi dice che è impossibile
non dovrebbe disturbare
chi ce la sta facendo"

- Albert Einstein -

Abstract

Mixed Reality, an evolution of Augmented Reality, is gaining a considerable spread during these days, in many application areas. In particular research is following many development directions in regards to medical therapy. Within this field, Mixed Reality is widely employed for surgical trainings, as for visualization of clinical folders or biological models; but it can also be used for distraction and pain reduction in pediatric hospitals, during a medical intervention or treatment.

The main goal of this thesis work is the development of a Mixed Reality application for children's pain reduction in pediatric hospitals. The designed software, which implements a voice-controlled video game, follows determinate boundaries posed by the specific application field and the limits of the adopted hardware.

Two specific Mixed Reality headsets, HoloLens and Magic Leap One, are taken into exam. Their features are discussed accurately and experimentally tested to reach a final comparison in both subjective and objective terms.

Sommario

La Realtà Mista, evoluzione della Realtà Aumentata sta avendo negli ultimi anni notevole sviluppo, in diversi ambiti di applicazione. Tra di essi l'ambito medico e terapeutico, dove la Realtà Mista è utilizzata in training chirurgici, come nella visualizzazione di cartelle cliniche o modelli biologici; ma può essere tuttavia utilizzata anche come strumento di distrazione, durante un intervento o una medicazione in ambito pediatrico.

Lo scopo di questa tesi è lo sviluppo di un'applicazione di Realtà Mista per l'intrattenimento dei bambini negli ospedali pediatrici. Il software sviluppato realizza un video gioco a controllo vocale seguendo alcune specifiche dipendenti dal particolare settore di applicazione e dai vincoli delle piattaforme hardware.

Verranno anche presi in esame due specifici visori per Realtà Mista, HoloLens e Magic Leap One. Di essi verranno discusse le caratteristiche, per giungere infine ad una comparazione sia di tipo soggettivo che di tipo oggettivo tra i due.

Contents

Listing of figures

Listing of tables

Listings

Acronyms

AR	Augmented Reality
MR	Mixed Reality
VR	Virtual Reality
CG	Computer Graphics
CGI	Computer-Generated-Imagery
ubicomp	Ubiquitous Computing
XR	Extended Reality
HMD	Head Mounted Display
HUD	Head Up Display
OST	Optical See Through
VST	Video See Through
LCD	Liquid Crystal Display
DoF	Degrees of Freedom
GPS	Global Positioning System
CPU	Central Processing Unit
GPU	Graphics Processing Unit
HPU	Holographic Prcessing Unit
RGB	Red Green Blue
IMU	Inertial Measurement Unit
IPD	Interpupillary Distance
USB	Universal Serial Bus

RAM	Random Access Memory
CUDA	Compute Unified Device Architecture
SoC	System-on-a-chip
ML1	Magic Leap One
VRD	Virtual Retinal Display
FoV	Field of View
OS	Operative System
IDE	Integrated Development Environment
UI	User Interface
SDK	Sofware Development Kit
FPS	Frames per Second

1
Introduction

The Mixed Reality (MR) concept draws its origins in the Early Sixties, despite the widespread development and consistent diffusion across the world in the recent last few years.
MR was derived both conceptually and historically from Virtual Reality (VR), which was ideated by Ivan Sutherland in the early sixties. His idea aimed in creating a sort of completely "Artificial Reality", which surrounds the users and interacts with them. [13]
However the first example of a real MR application was "Sensorama", a video arcade invented by Morton Heiling in 1962, which could emanate odors and transmit vibrations to the user. Although, it was not interactive yet. [14] [1]
Afterwards, Ivan Sutherland continued Heiling's work at the Utah University, and in 1968 developed the first VR Headset [fig. 1.1(a)]. Then in the 1970s thanks to the diffusion of Computer Graphics in simulations and the improvements of the computing architectures, MR technologies started upgrading to reach the spread and fame they have today.

Since its very beginning, the success and diffusion of MR applications could be easily forecasted since they proved to be well-suited for many different applications, in multiple areas. Medical theraphy is one of the most important to mention in order to emphasize its relevance.
In 1994, the University of North Carolina at Chapel Hill presented a compelling medical Augmented Reality (AR) application, capable of letting a physician observe a fetus directly within a pregnant patient [fig. 1.1(b)]. [1]

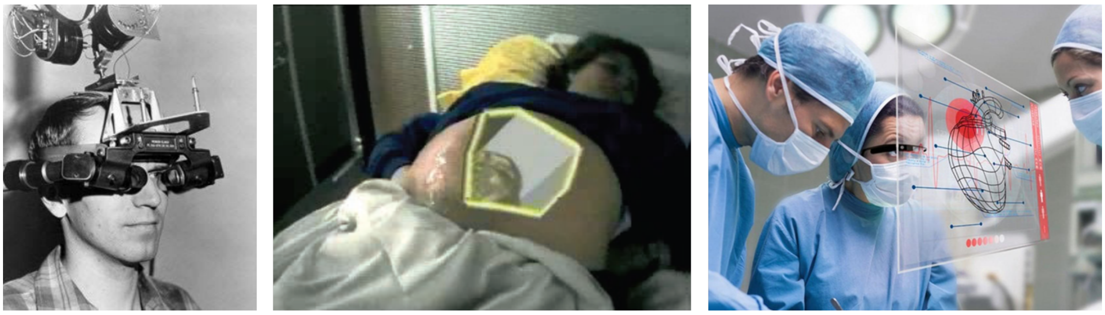


Figure 1.1: (a) The sword of Damocles was the name of the first HMD, build in 1968. (b) View inside the womb of an expecting mother. [1] (c) Medical imaging nowadays

Nowadays a considerable number of societies are working in the field; Between them, the "Dreamship studios" of San Francisco (California), founded by Hannah Luxenberg[1], focuses on designing MR technologies for medical applications.
As Kodak asserted

> Hannah Luxenberg is helping to build the future for children's hospitals and hospices. Working closely with patients, doctors, and research teams to develop the next therapeutic technology programs, Hannah effectively envelops patients in calming and engaging therapeutic scenery, providing an alternative means of therapy and a method for coping with emotional and physical pain through the power of virtual reality.

In addition to it, many important societies related to the medical fields are employing MR systems for surgery, rehabilitation and medical imaging. It is worth to be mentioned "The Chariot Program" by Dr. Samuel Rodriguez at Stanford's Lucile Packard's Children's Hospital[2], an innovative program that aims at making less stressful the experience of hospitalized children.
The goal of the program is to implement innovative technologies in the hospitals that prove to be familiar to children in an effort to engage and distract them during medical operations. The project developed in this thesis follows this purpose.
Our application, named "MR Dance", is a game for medical therapy applications, developed following the above discussed guidelines and constraints. The choice we made was to deploy it on three different platforms and with different input methods, to show how MR can fit

[1] More information on http://hannahluxenberg.com
[2] https://www.stanfordchildrens.org/en/innovation/chariot

well each one despite their different features and to compare their performances holding the same load.

We begin the thesis introducing the environment of Mixed Reality and the technologies involved, explaining then the behavior and characteristics of the most important Mixed Reality platforms; finally we describe the project and provide the results obtained from the tests performed.

2

Overview on Mixed Reality world

In this chapter an overview of Mixed Reality world is given and it is compared to the most common related technologies in its field.

2.1 Definitions and scopes

The most popular definition of Augmented Reality was proposed by Ronald T. Azuma in his 1997 survey paper. [15] According to him, AR systems must have the following three characteristics:

- they combine real and virtual;
- they must be interactive in real time;
- they are registered in 3D.

This definition is sufficiently general to be applied to a large number of devices, including Head Mounted Displays (HMD) (see section 3.1) and common smartphones.
But then, what is the real difference between Augmented Reality, Mixed Reality and the other related similar technologies? We are going to examine each of them in the following section (2.2).

The most general distinction between Augmented/Mixed Reality and Virtual Reality is that AR allows the user to see the real world, with virtual objects superimposed upon or composed with it; AR supplements reality, rather than completely replacing it. [1]. Conversely, Virtual Reality totally immerses the user in a synthetic computer-generated world.

2.2 Mixed Reality continuum and related fields

Broadly speaking, it is possible to refer to technologies that merge real and virtual worlds, using the term Mixed Reality[1].

As we can perceive synthetic scenes on various levels of virtualization, it is possible to highlight the peculiarities among different MR technologies, referring to the quantity of real and virtual elements present in the scene. According to Paul Milgram's paper of 1994 [16] [17], we can define a taxonomy of the various ways in which "virtual" and "real" aspects of MR environments can be arranged.

To realize this purpose, the concept of "Reality-Virtuality continuum" has to be introduced [fig. 2.1]. It relates to the mixture of classes of objects presented in any particular setting where real environments are placed at one end, and virtual environments at the opposite end.

The former (on the left), refers to environments consisting solely of real objects, and display the processing results using standard displays. The latter (on the right), defines environments consisting solely of virtual objects, e.g. a conventional computer graphic simulation.

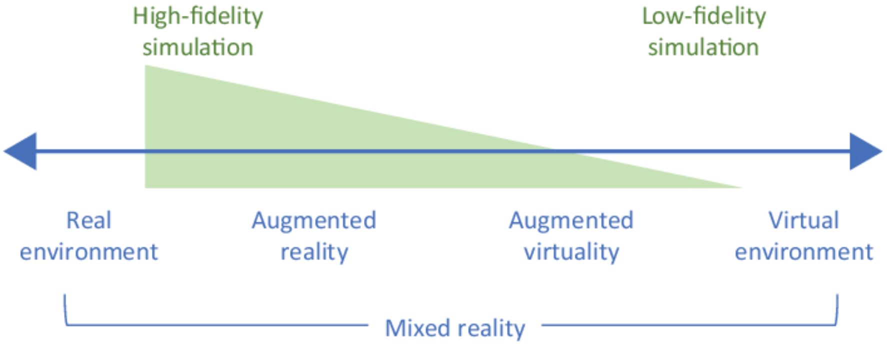


Figure 2.1: Reality-Virtuality continuum [2]

[1] The term was introduced by Steve Mann (section 2.2.3) of Mannlab https://mannlab.com

2.2.1 Augmented Reality

The definition of "Augmented Reality" given in section 2.1 is the most general one and can be associated to the whole continuum spectrum defined in section 2.2.
It can be considered very close to Mixed Reality since they both include the real environment and computer-generated images (CGI) to have for result a mixture of them.
The difference between MR versus AR is mostly semantics but deeply critical to some people working in the field. Thus Augmented Reality and Mixed Reality can be distincted considering the level of integration between real and virtual elements. [18]
An Augmented Reality appication superimposes CGI elements over real ones, but avoids every kind of interaction. Mixed Reality includes all the elements, virtual and real, homogeneously, since it has an accurate knowledge of the surrounding world and where things are.

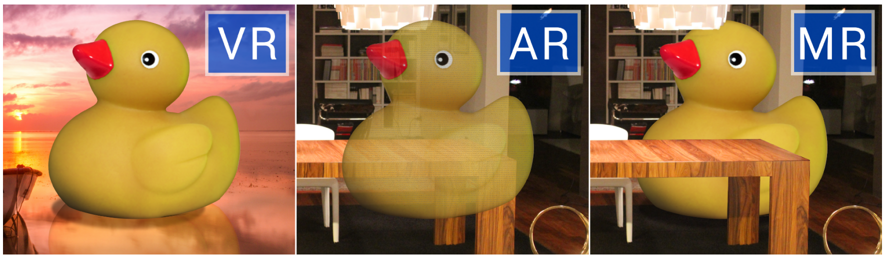


Figure 2.2: Visual differences between VR, AR and MR [3]

2.2.2 Virtual Reality

As the continuum suggests, VR environment is completely computer-generated, so it has no reference to the real world and involves all of the user's senses, isolating them from reality.
Since the very beginning of virtual reality technologies, several moral and ethical debates arise.
Some people sustain that the artificial elements synthesized by VR systems, so realistic, could induce the user to lose the sense of space and time, and even the contact with reality entirely.
The technologies cited above do share some basic features, but offer totally dissimilar experiences. Also target applications and related headsets are often different. VR is becoming increasingly popular in computer games industries, while AR and MR are employed in a larger number of sectors, like Medicine, Games, Navigation, Advertisement and so on. VR visors are several; the most common on commerce are Oculus Rift and HTC Vive, but there

exist also visors suitable both for AR and VR. MR specific devices are treated in chapter 3. [1] [18]

2.2.3 Ubiquitous Computing

In 1991 Mark Weiser proposed the concept of "Ubiquitous Computing" (ubicomp) in his seminary essay [19] and wrote that

> the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.

His work discussed the diffusion of digital technology in our everyday life as a mean to support it. This possibility is, step by step, reaching such a high level of integration with our everyday lives that nowadays we can not think them separately any more.
Recent advances in sensors, embedded microsystems and wireless communications, have brought to the constantly-increasing diffusion of distributed computing platforms, described by Weiser twenty-eight years ago.
Ubiquitous computing moves processing and communication technology beyond the personal computer to everyday devices enabling the access to digital data anytime and anywhere. An example of such devices are smartphones, driver assistance systems in cars, home automations and the body area networks. Mark Weiser was predicting the concept of "Internet of things".

On the other hand, Ubiquitous Computing lies on the opposite of Virtual Reality. In fact Weiser stated that ubicomp combines "Virtuality" and "Ubiquity" rather than isolating the user apart from the real world as VR does; the real location and place are considered computational inputs.
Therefore "Ubiquity" describes the degree to which information access is independent from being in a fixed place (a terminal). From these consideration it is possible to rearrange the Milgram's continuum into a 2D diagram, extended by Weiser's statements and consequently taking the name of "Milgram-Weiser continuum" [fig. 2.3]. [20]

2.3 MR philosophy

In order to be respectable and standard, Mixed Reality technologies have to be ruled. Particularly, they need to have defined boundaries as regards the ethical level.

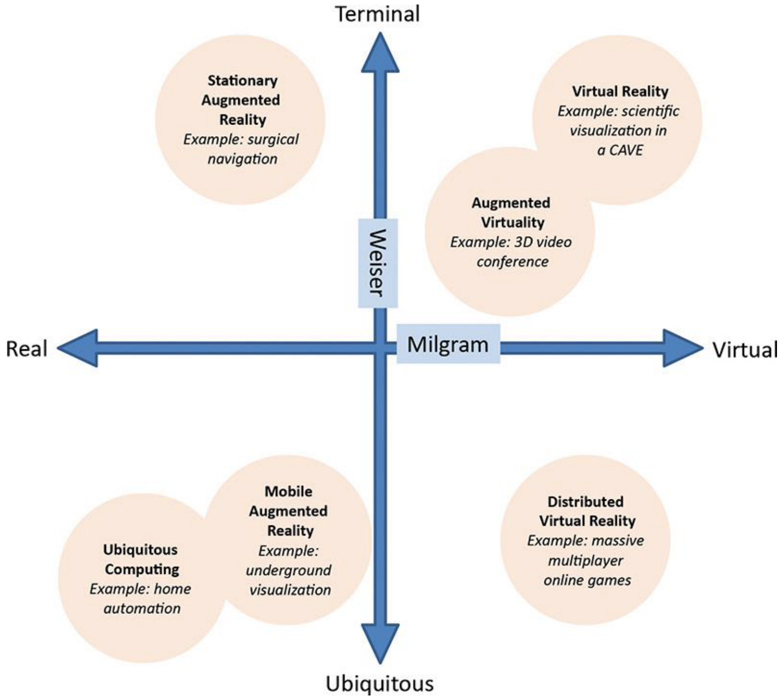

Terminal
Stationary Augmented Reality
Example: surgical navigation
Virtual Reality
Example: scientific visualization in a CAVE
Augmented Virtuality
Example: 3D video conference
Weiser
Real
Milgram
Virtual
Mobile Augmented Reality
Example: underground visualization
Ubiquitous Computing
Example: home automation
Distributed Virtual Reality
Example: massive multiplayer online games
Ubiquitous

Figure 2.3: The Milgram-Weiser chart, relationship of various user interface paradigms [1]

The first step towards this regulation was in 2004 when Steve Mann introduced the "Code of Ethics on Human Augmentation" in his Keynote Address at Transvision. [21] [22] This code sustain that

> As we augment our bodies and our societies with ever more pervasive and possibly invasive sensing, computation, and communication, there comes a point when we ourselves become these technologies.

In addition, the document defined three fundamental laws, representing a set of guidelines towards a philosophical ideal (like the laws of physics, or like Asimov's Laws of Robotics[2]), rather than an enforcement paradigm. [18]

[2] The Three Laws of Robotics, from 'Asimov, Isaac (1950). "Runaround". I, Robot':
1) A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2) A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3) A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.

In 2016, John Rousseau proposed three laws to ensure that augmented and virtual technology positively impacts society. Rousseau said that

> The future of human consciousness will be a hybrid affair. We will live and work in a ubiquitous computing environment, where physical reality and a pervasive digital layer mix seamlessly according to the logic of software and the richness of highly contextual data. This is mixed reality.

Rousseau, citing Isaac Asimov's "Laws of Robotics" suggested these three "Laws of Mixed Reality" that will help us shape the discourse and future development of Mixed Reality with an emphasis on preferable outcomes. The laws are aligned to three significant problem areas, covering the individual, society and economics.[23]

1. Mixed Reality must enhance our capacity for mindful attention.
2. Mixed Reality must embody a shared human experience.
3. Mixed Reality must respect boundaries between commerce and data.

2.4 Examples and development

In Industrial applications, Mixed Reality technologies can be used to create an interactive application that evaluate the placement of new machinery or workstations inside an existing manufacturing plant by visualizing the future plant on the real images. The main advantage is the possibility of evaluating in advance the suitability of the installation by visual inspection, determining whether the new tools are or not in conflict with older ones.

In addition, several MR applications have also been conceived in the military fields.
Armed forces has played an important role in the creation and development of wearable augmented reality, starting back in 1963 with Bell helicopter which inspired Ivan Sutherland, the father of Computer Graphics (CG).

MR is also widely used in design and advertising, as a tool for design review and pre-evaluation of the ongoing models while design is still in the development stage. Virtual models that replace real ones could be used to inform customers and public about new products.

The medical field is so broad that there are dozens of applications (See chapter 1); from ensuring mobility to health care workers in a hospital or medical's office and granting access to their health records, to allow a surgeon from a remote hospital across the world to assist and coordinate an operation at a different location [18]. MR also improves the way healthcare professionals are trained and educated.
The project treated in this thesis (chapter 4) can related to the medical field, since it is finalized to make the children more comfortable with a medical approach.

Figure 2.4: Examples of MR applications in Industry, Design, Medicine and Military

2.4.1 XR devices

XR stands for "Extended Reality", and it is defined as

> Technology-mediated experiences that combine virtual and real-world environments and realities. Here the 'X' can be seen as a placeholder for V(R), A(R) or M(R), though it also represents an undefined or variable quality/quantity. Most definitions of XR encompass platforms and content where the user can take digital objects into reality, or, conversely, see physical objects as present in a digital scene.

This definition is the one given in the Unity 3D's Glossary [12].
Unity3D is the development software used for the project described in chapter 4. It is a

cross platform game engine first released in June 2005 as a Mac-OS X-exclusive game engine and then extended to support more than 25 platforms. The main use of the engine is to create three-dimensional, two-dimensional and mixed reality games, as well as simulations and other experiences; despite this initial use, the engine has been adopted by industries outside video gaming, for the realization of films, architectures and so on.
Section 4.1 will describe Unity 3D in more detail. Here we highlight that Unity enables to export projects to different target devices and platforms (See figure 2.5).

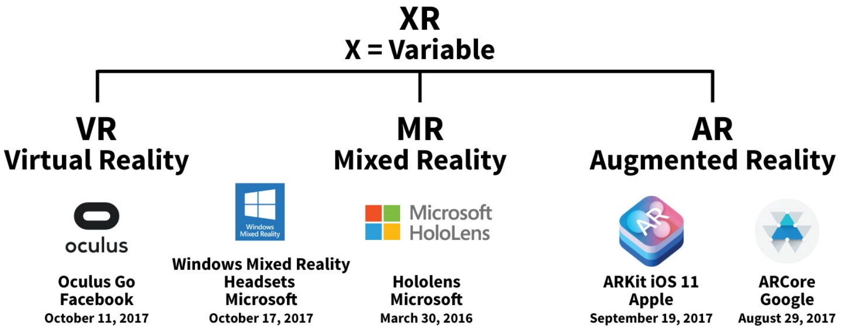


Figure 2.5: XR main supported devices [4]

Go as far as you can see; when you get there,
you'll be able to see farther.

- J. P. Morgan -

3 HoloLens and Magic Leap One

In this chapter the behavior of MR visors were examined, focusing in particular on two of them: Hololens and Magic Leap One.

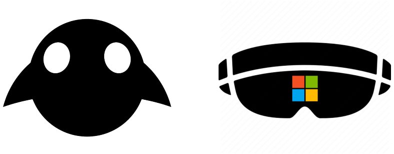


Figure 3.1: Magic Leap and Hololens logos [5]

3.1 A Taxonomy of MR devices

Already in 1994 Paul Milgram stated that the different devices for MR applications needed to be categorized through a taxonomy. To this purpose he wrote an essay [17], where the different systems were classified in several ways.
A simple inital classification divides devices into wearable and not; then a further separation could be performed considering their structural characteristics as we can see in figure 3.2.

3.1.1 Wearable

The "Wearable" devices, also known as Head Mounted Displays, are totally dedicated to AR/MR. Into this category we can find helmets and headsets.

Helmets A device is classified as a helmet if it covers the user's ears, total head and most of the face. An example is the Jarvish's motorbike helmet[1], which can visualize information about the vehicle directly on the frontal glass during the driving.

Headsets A device is classified as a headset if it is less invasive and its dimension is smaller compared to a helmet. The most of actual devices on the market like smart glasses and visors are headset. Between them we can find HoloLens and Magic Leap One, accurately described in sections 3.3 and 3.4.

3.1.2 Non wearable

The "Non wearable" devices are also "non dedicated", because they are common devices which can perform the function of MR supporters. As a matter of facts this category comprehend mobile devices, like smartphones and tablets, and stationary devices, like TV or PC. Head-up Displays (HUDs) devices deserve a more careful examination reported in the following paragraph.

Head-up Displays These devices are mostly-used in military aircrafts and vehicles, which allow the user to examine the displayed data without staring away from the frontal screen. It also has the advantage that the pilot's eyes do not need to refocus.
As in all categorizations, classes can sometimes overlap since these are simple generalizations. [18] [24]

3.2 Base principles of MR devices

Mixed Reality is evolving into multiple directions, since the huge number of technical requirements have been solved into several ways, with different specific features.
Most of these technologies need to localize the objects and the location and the orientation of

[1] Further information on https://www.jarvish.com

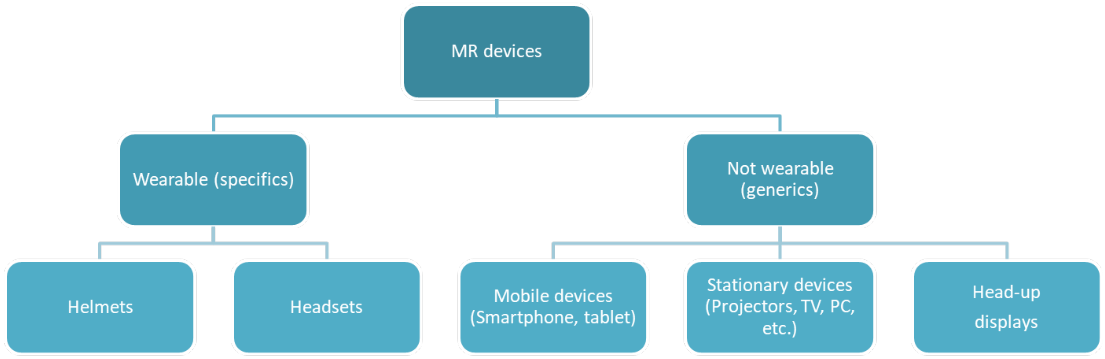


Figure 3.2: Taxonomy of Mixed Reality devices

the display (6 degrees of freedom - see section 3.2.2). In some situations, the tracking system can be physically attached to the display. [13]

3.2.1 Display

Regarding the visualization of contents, we can group displays in three/four macrocategories as we can see in figure 3.3.
All of these classes of displays share the common feature of juxtaposing "real" entities together with "virtual" ones; however Paul Milgram asserted that is necessary to make a distinction, because

> Some systems are primarily video based and enhanced by computer graphics, others are primarily computer graphic based and enhanced by video; in some systems the real world is viewed directly (through air or glass), whereas in others real-world objects are scanned and then resynthesised on a display device (e.g. analogue or digital video). [17]

3.2.1.1 Optical characteristics

Optics technology is at the core of defining the MR experience delivered by any HMD. In fact the future adoption of MR smartglasses will likely be fostered by the evolution of optical components, their performance, and their cost.
We can distinguish two types of visualization for MR systems:

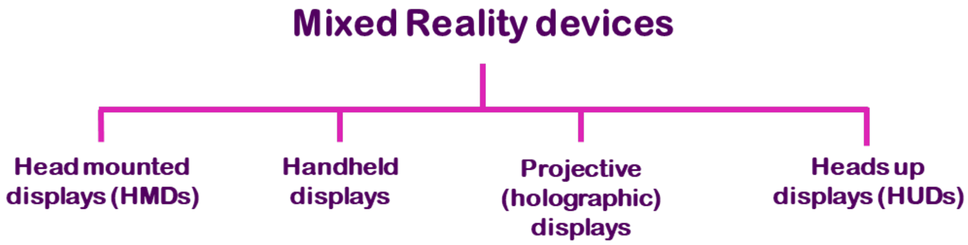


Figure 3.3: Categorization of MR displays' according to the technologies employed

Optical see through In Optical see through devices, the user perceives the real world and the virtual objects through an optical combiner (a mirror), which is partially transmissive and partially reflective. The mirror lets a sufficient amount of light from the real world passing through to see it directly, and at the same time it displays Computer-Generated-Images overlaid onto the real world [fig. 3.4(a)]. Microsoft's HoloLens, Magic Leap One and the Google Glass are recent examples of optical see through HMDs.

Video see through In the devices exploiting this principle, the user views reality that is first captured by one or two cameras mounted on the display, so the combination of virtual and real is realized by the graphic processor of the computer and not by the user. The graphic processor simply blends the video image taken from the cameras and the Computer-Generated-Images and hands the final blend to the user [fig. 3.4(b)]. The HTC Vive VR headset has an inbuilt camera which is often used for creating AR experiences on the device. [1] [25]

In contrast with that we have just seen, VR goggles do not need a see-through capability and therefore it presents fewer optical challenges than MR/AR devices.
In addition there is another important principle of functioning, which was pointed out by Azuma in his symposium in 2018[2]. We are talking about Projective MR.

Projective MR It is the most popularized way to do MR, presented in the Star Wars series, Minority report and the Iron Man series in the recent time. These type of displays use light diffraction to generate three dimensional forms of objects in real space.

[2] http://www.imaging.org/Site/IST/Conferences/EI/Symposium_Overview.aspx?New_ContentCollectionOrganizerCommon=2

In this approach, the desired virtual information is projected directly on the physical objects to be augmented. This is a sort of optical combination, where we don't need any display or optical combiner. It is also known as "Volumetric 3D display" for which the points of light defining perceivable objects are physically distributed throughout a 3D volume. [fig. 3.4(c)] [24] [1]

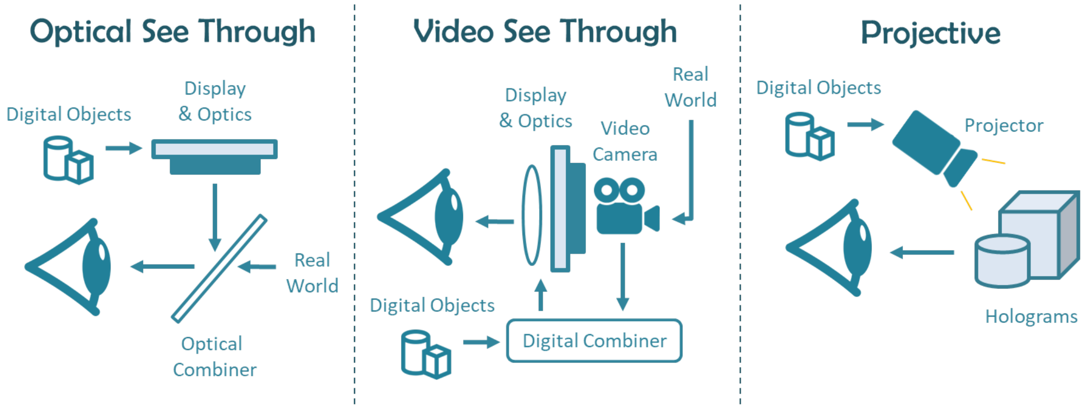


Figure 3.4: The three types of HMD displays' visualization

3.2.1.2 Further categorizations

The macro grouping we allude in section 3.2.1 and in figure 3.3 is the following:

Head Mounted Displays This category includes the "Wearable" devices seen in section 3.1.1. As we talk about head-worn displays, or smartglasses integrated with lenses and other functional elements (such as a microphone, camera, or earphones), we are referring to the same thing. [18] They are probably the most common type of displays used in MR and consist of one or two visual display units together with optically compensated systems that form a perspectively correct virtual image, even though the display is very close to the user's eyes. They are based on the principle of "See through", "Video see through" or "Optical see through" depending on the construction technologies employed. HoloLens and Magic Leap One own to this category and make use of the "Optical see through" principle.

Handheld displays Handheld displays are flat panel LCD displays that use an attached camera to provide "video see through" based augmentations. The handheld display acts as a window or a magnifying glass that shows the real scene enriched by virtual elements. Handheld displays are normally less expensive than HMDs as there is no need for optical compensation. Smartphones and tablets belong to this category.

Projections Displays Rather than addressing a user's perception through a display, like an HMD or handheld, an alternative is to project computer generated images directly onto the environment using standard video projectors exploiting the principle of "Projective MR" explained in section 3.2.1.1.

The last category, called head-up displays, is described in section 3.1.2. A regular HUD contains a projector unit, a viewing glass and a computer. It can also be integrated into a HMD, so it bases on "Projective MR" and/or "Optical See Through" principles. [13] [24]
As we can see in figure 3.5 usually the technique for visualization employed depends on the proximity of the display medium in respect to the user.

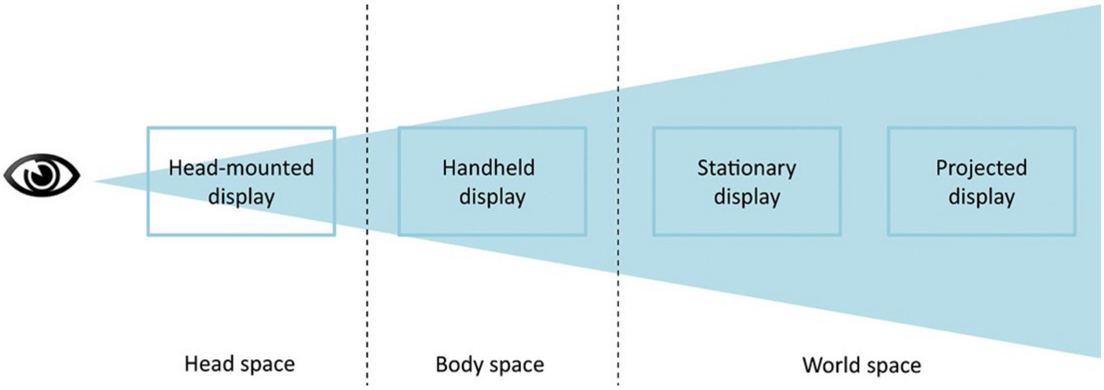


Figure 3.5: Categorization of MR displays' according to the distance from eye [1]

3.2.2 Tracking

Tracking refers to the determination of position and orientation of the viewer at runtime, which require to be measured continuously in order to ensure a smooth viewing experience. In fact, to correctly display virtual objects in the 3D scene, it is necessary to be aware at least of their "relative pose", which is the position and orientation of the MR display in respect to the real world's objects. Tracking systems use 6 degrees of freedom or "DoF", 3 degrees for location in the 3D space and 3 degrees for rotation [figure 3.6].

Tracking must not be confused with other two important related properties of MR systems: registration and calibration, whose terms overlap in practical use. Their meaning is clarified in the following paragraphs.

Calibration It is the process of comparing measurements coming from two different devices: a reference device or a known coordinate system, and a device to be calibrated. It is usually carried out at discrete time, unlike tracking. It is also employed on the devices used for tracking and it can be done once or many time during a session.

Registration Recalling the definition given in section 2.1, registration refers to the alignment of coordinates systems between virtual and real objects, after estimating the poses of user's head or/and of the camera. Specifically see-through displays should show CGI elements such that they align with real-world objects using registration.
Registration can be static or dynamic depending on whether the user's head or the camera is moving or not. Static registration can be performed with high accuracy using calibration, while dynamic one requires fast tracking algorithms. [1]

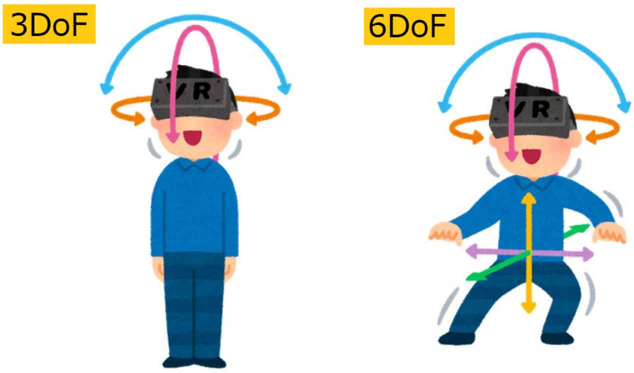


Figure 3.6: Degrees of freedom. (a) Only rotation. 3 degrees of freedom - 3DoF (b) location and rotation. 6 degrees of freedom - 6DoF [6]

3.2.2.1 Features of tracking systems

We have just discussed that registration needs a coordinate system to align virtual and real objects (section 3.2.2). This is possible because MR relies on a standard computer graphics pipeline to produce overlays on the real world, which consists in three steps: a model transformation, a view transformation and a projective transformation. [26]

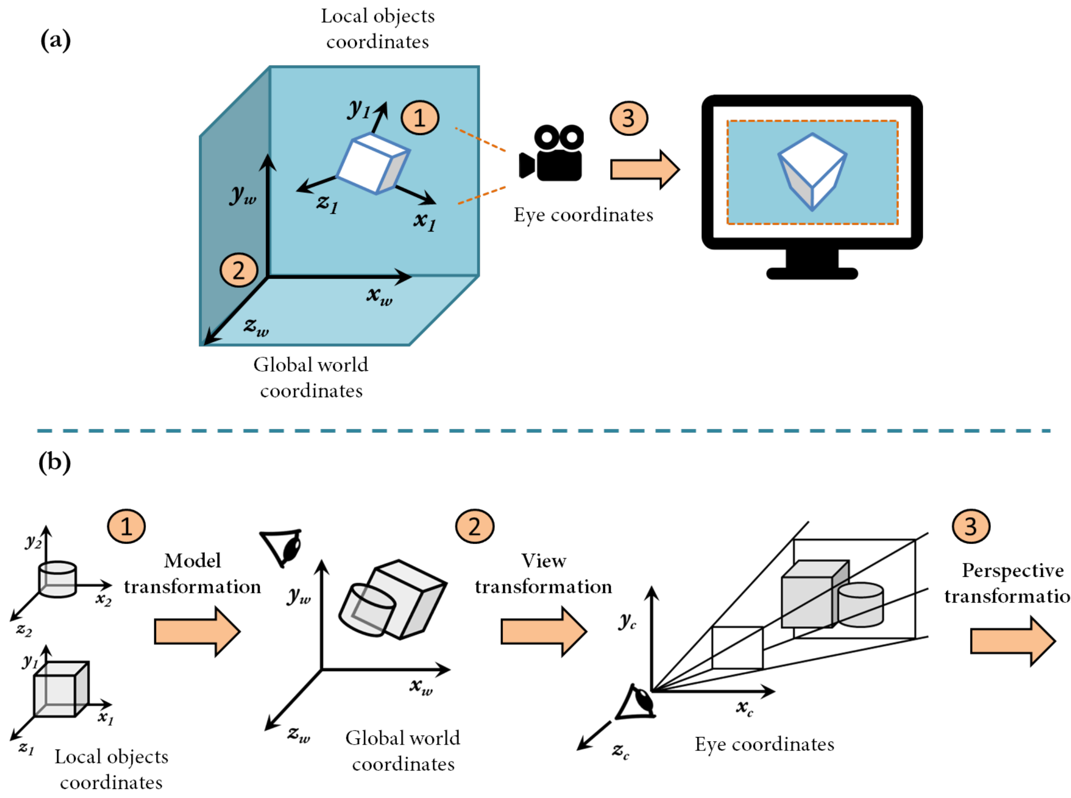


Figure 3.7: MR coordinate system. (a) Full scene (b) Detail. 3D geometry pipeline

Each transformation corresponds to a different MR coordinate system. The model transformation describes the pose of the moving objects into a static environment, which can be defined by geographic coordinates, a feed from a server on the web, or the user's interaction; the view transformation describes the pose of the camera/display obtained by a tracking sensor, visual features or a geo-location system; the perspective transformation describes the mapping from eye coordinates to screen coordinates. Both the model and the view transformations provide the alignment of objects and can be tracked enabling registration. [fig. 3.7]

Measurement systems used in tracking technologies can employ a variety of physical phenomena and arrangement options, that determine which coordinate systems are being used and influence the spatial and temporal properties of the tracking. [1]
Defining the position and orientation of a real object in space is determined with the help of special sensors, which record the signal from the real object when it moves or is moved

and transmit the received information to the computer. Examples of these sensors are magnetometers, accelerometers and gyroscopes.

3.2.2.2 Types of tracking

Tracking systems are of many different types and are widely used for purposes different from MR. The most common tracking systems used within MR environments own to the category of "optical tracking", which can be subdivide into two sub-categories, "inside-out" and "outside-in" tracking systems, depending on the position of the sources and sensors. In inside-out positional tracking, the camera or sensors are located on the device being tracked (e.g. HMD) while in outside-in the sensors are placed in a stationary location [fig. 3.8]. HoloLens (section 3.3) relies on an inside-out tracking system, like most of VR devices, while outside-in is used by HTC Vive.
The leading principles on which Optical Tracking Systems for MR are based, are presented below.

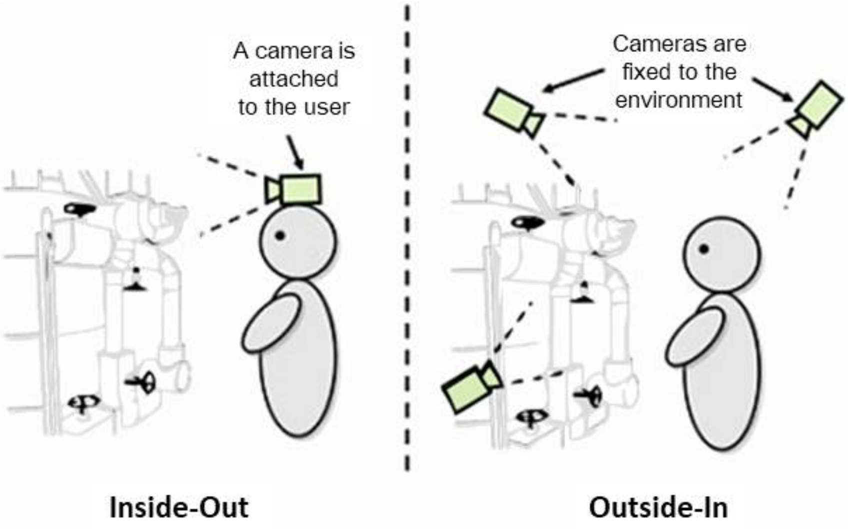

A camera is attached to the user
Cameras are fixed to the environment
Inside-Out
Outside-In

Figure 3.8: Differences between Outside-in and Inside-out approaches [7]

Visual Markers Some tracking systems employ a set of visual markers, i.e. a set of image patches that can be specifically designed to track the movements of the device/HMD. These features are used in designed in combination with a computer vision recognition algorithm

to yield high probability of recognition and low probability of misclassification. After detecting the markers, an optical flow algorithm compute the relative motion of the viewer, and computed the new pose place virtual 3D objects in the real world.
Within the realm of MR, marker systems are used to enable real-time camera pose estimation with high accuracy and low latency [3].
We can use also other target objects in place of Visual markers and the best visualization is achieved using video see-through displays (section 3.2.1.1). [27] [13]

Markerless tracking Alternatively we can use a different approach, which does not need the placement of markers or target objects and allow to compute the visual markers directly from the captured scene or from depth sensors. In this case, computer vision techniques recognize and track distinctive features such as faces or objects with specific textures or contours. The most frequently used markers are called “keypoints” and consists in the local salient points of a target object. As an example the keypoints for a cube to be localized will be its visible vertices.
These systems normally require a training phase in which the objects to be tracked are presented to the system from one or more viewpoint angles and then data are stored into a database. Despite marker-less systems do not require the placement of extra objects, they are computationally expensive, have lower accuracy (high risk of misclassification) and higher latency than marker based. [1] [13]

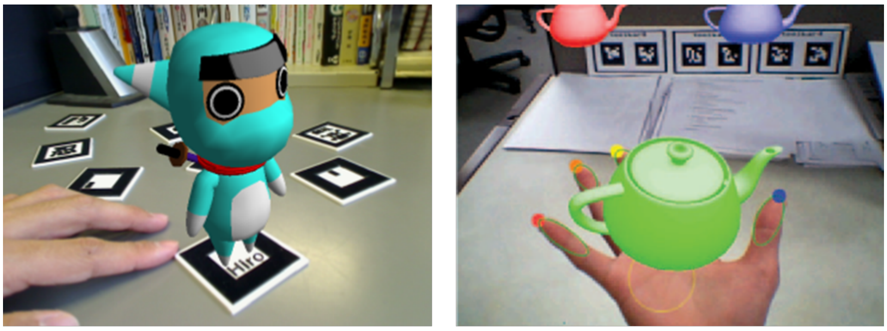
Furthermore we can mention other popular types of tracking, not optical based. A clear example is the Global Positioning System (GPS), which exploits radio signals continuously broadcasted by a number of medium earth orbit satellites to calculate the location of the receiver.
Other systems rely on magnetic or acoustic properties for tracking. Respectively, the former makes use of ultra sound waves and the latter of a fixed transmitter and some sensors, for measuring the position and orientation of target objects. Another example is inertial tracking, using gyroscopes and accelerometers.
Often, many devices resort to hybrid systems, neither active nor passive, which employ a mixture of tracking technologies taken from two or more tracking systems, e.g. inertial and

[3] Accuracy is a measure of the error in the position and orientation reported by the tracker.
Latency is the delay between the change of the position and orientation of the target being tracked and the report of the change to the computer

optical tracking [4] or magnetic and vision–based [5].
As reported in “Hybrid tracking for augmented reality” [28]

> The fusion of complementary sensors should be used to build better tracking systems, synergies can be exploited to gain robustness, tracking speed and accuracy, and to reduce jitter and noise.

Nowadays, a hybrid tracking system, even though its complexity, is the best strategy to achieve a better pose estimation. Among these solutions, we can mention the marker based system produced by TACTIC[6], which collaborates in this project.



Marker based tracking

Markerless tracking

Figure 3.9: Marker based tracking vs Markerless ones [8] [9]

3.3 HoloLens

Microsoft HoloLens is a device for Augmented and Mixed Reality, developed and manufactured by Microsoft. It was released on the market on 30th March 2016 and it follows the lineage of Kinect, an add-on for Microsoft’s Xbox gaming console that was introduced in 2010. [29]
Its price is 3000$, a little expensive to permit a large scale diffusion, but sufficiently low to gain popularity and wide diffusion especially in the research field. As a matter of fact

[4] R. Azuma and G. Bishop, “Improving static and dynamic registration in an optical see-through hmd,” in SIGGRAPH, 1994.

[5] T. Auer and A. Pinz, “Building a hybrid tracking system: Integration of optical and magnetic tracking,” in IWAR, 1999.

[6] https://tactic.studio

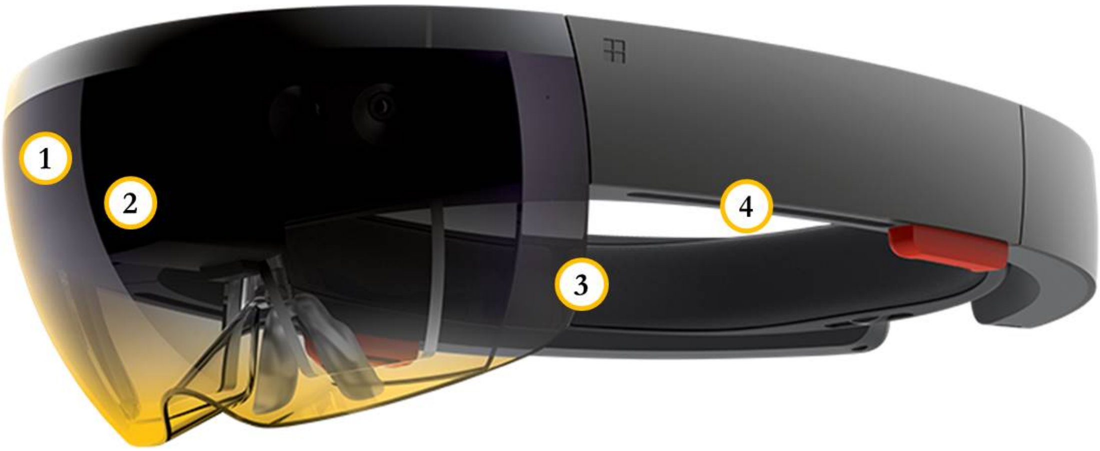


Figure 3.10: Hololens. (1) Cameras and Sensors (2) Integrated Computer (3) Optical Lenses (4) Spatial Sound

HoloLens offers a variety of development products grouped under the name of "Windows Mixed Reality", a complete platform based on the Windows 10 operative system to develop Mixed Reality applications.
On 24th February 2019 the new version of HoloLens was announced at the Mobile World Congress in Barcelona, Spain and for now it is disponible in preorder at the price of 3500$.

3.3.1 Structure

From a structural point of view, HoloLens is a Head Mounted display with a Headset structure (section 3.1.1) and bases on the optical see through principle (section 3.2.1.1). It is entirely powered by a rechargeable battery with a life approximatively of 3 hours.
It is composed of many sensors and cameras, a pair of lenses and an integrated computer, so it is not very slight, but his weight (579 g) is well distributed and it is comfortable to wear. [fig. 3.10]

Cameras and Sensors The most of the sensors and related hardware including the cameras and processors are located in the front side of the visor. These cameras and sensors allows to reconstruct the depth and the environment of the surrounding ambient, take from it the light and adapt the virtual images juxtaposing on it. [fig. 3.11(1)] They allow a field of view of about 35° horizontally and 18° vertically.

Integrated Computer Being HoloLens self-stand, it is equipped with a powerful internal hardware, a CPU, a GPU and a Custom Build Holographic Processing Unit (HPU). The internal hardware has got a RAM memory of 2GB and a Flash memory of 64GB dedicated. As regards CPU, it is a 64-bit Intel-Atom x5-Z8100 with 1.04 GHz dedicated, while the GPU covers 114MB for Video Memory and 980MB for shared system. [fig. 3.11(2)]

Optical lenses HoloLens has a pair of transparent combiner lenses, where the projected images are displayed in the lower half. They exploit the principle of optical see through as anticipated and are composed of three layers for the primary colors (RGB). The visualization of virtual images through the lenses is possible thanks to other important components, the light engines and the inertial measurement unit (IMU).
The light engines is responsible to gather light's particles and project virtual images on lenses, additionally making a calibration to the interpupillary distance (IPD), or accustoming vision of the user.
The IMU instead, groups the tracking sensors (magnetometers, accelerometers and gyroscopes) and quickly transmits the collected data to the HPU. [fig. 3.11(3)]

Spatial Sound Finally, located near the user's ears, are a pair of small, red 3D audio speakers. The speakers do not obstruct external sounds and allow the user to feel an immersive audio experience without headphones.
Using head-related transfer functions, a specific model which characterizes how the human head receive sounds from a location, the HoloLens generates audio simulating spatial effects, so the user virtually perceive a sound as coming from a determinate position in the real world. [fig. 3.11(4)]

Along the visor there are on the top edge two pairs of buttons, shaped differently to be distinguishable by touch, for display brightness and volume; then at the end of left wing a power button and row of five, small individual LED nodes, used to indicate system status and finally on the bottom edge is placed a USB 2.0 micro-B to connect the device with a computer or to the power supply. [10]

3.3.2 Interface

Through the use of the HPU, HoloLens allows the user to interact with the scene by means of some natural interface commands, using their voice, hands or head.

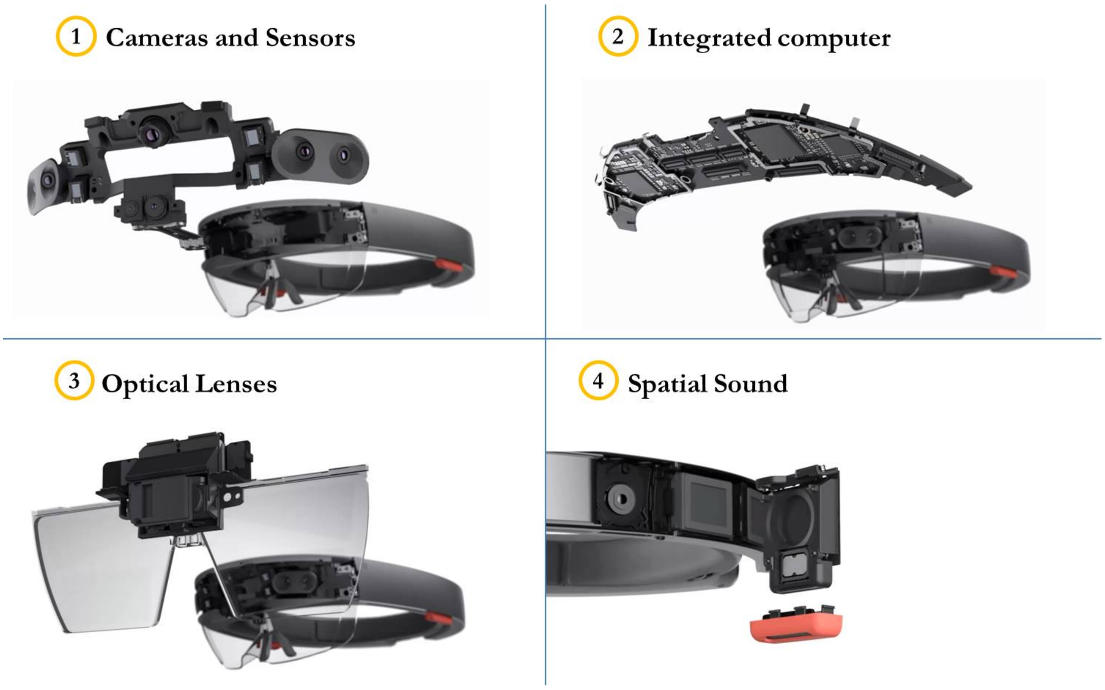


Figure 3.11: Hololens's Modules [10]

Gaze Gaze commands, such as head-tracking, bases on the sensors data to allow the user to bring application focus to whatever they are perceiving. Thanks to Gaze commands, HoloLens can create a virtual cursor to make the virtual objects selectable and interactive for the user. We underline that Gaze regards head-tracking and does not support the tracking of eyes' motion.

Gestures There are many types of gestures that HoloLens is able to recognize. "Air tap", a gesture similar to clicking an imaginary computer mouse, permits selecting an element; it can be hold and dragged whenever the user wants to navigate a menu or move an element. A "bloom" gesture for accessing the shell can be performed by opening one's hand, fingers spread with the palm facing up. [fig. 3.12(b)]
Windows can be dragged to a particular position, as well as resized. Virtual elements such as windows or menus can be "pinned" to locations, physical structures or objects within the environment; or can be "carried", or fixed in relation to the user, following the user as they move around.

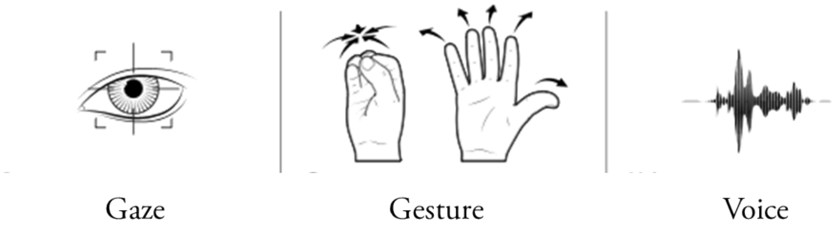


Figure 3.12: Hololens's commands

Voice Many commands can be executed also by means of voice input. HoloLens has a skill of speech recognizer and is able to transform user's pronounced words into executable commands. This method will be employed for the project in section 4.3.2.

3.3.3 Uses

The visor covers the whole Virtuality-Reality spectrum (section 2.2).
On one hand it can show virtual generated or recreated environment or simulated physical worlds. There the user can navigate and interact with objects, but he can also move a real one by means of the simulation, such as a rover on the Mars's surface.
On the other hand HoloLens is more used for Mixed and Augmented reality, to add some virtual elements to the real world. It provides a high-fidelity spatial mapping of the real surrounding world, acquiring data about the surfaces and the materials of which it is composed (See section 4.3.2).
The virtual object are called "Holograms" and thanks to the mapping, the user can walk around them, without let them moving with his head.

3.4 Magic Leap One

Magic Leap One is an HMD for MR/AR, developed by the tech startup Magic Leap founded by Rony Abovitz in 2011, and firstly appeared on the market on 6th June 2018. It is the first serious competitor of Microsoft HoloLens and still has a high price, 2295$, which is inaccessible to the majority of ordinary users.

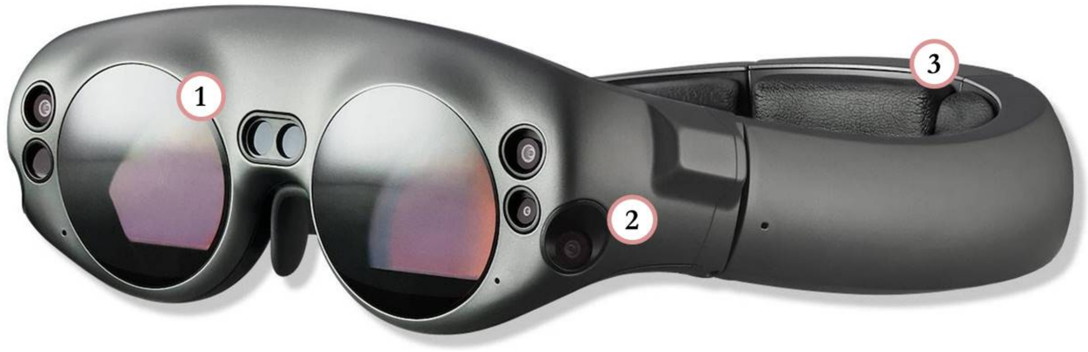


Figure 3.13: Magic Leap One. (1) Optical Lenses (2) Cameras and Sensors (3) Spatial Sound

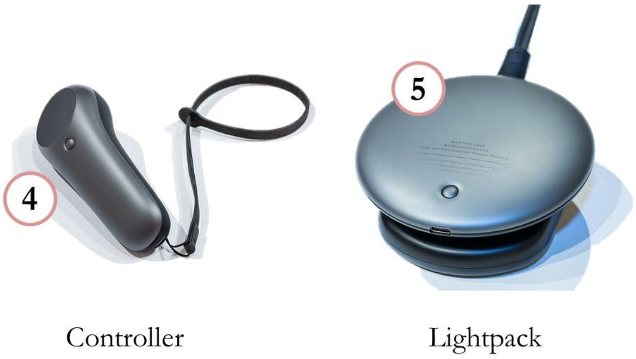


Figure 3.14: Complementary devices for Magic Leap One's visor [fig. 3.13] (4) Controller (5) Lightpack

3.4.1 Structure

Magic Leap One consists of a computer named "lightpack", a headset named "lightwear" and a controller [fig.3.14].
The headset is lightweight and comfortable, and also quite pretty for its fancy rounded lenses. It combines in itself several built-in components: sensors, cameras, two lenses, a 3D audio system and a microphone.
The lightpack carries the central and graphic processors, namely the bulk of Magic Leap One's hardware, so it is the heaviest object of the bunch (415 g) and lets the visor to be lighter on the user's head (345 g). According to the instructions it has to be carry on the belt. Like HoloLens, Magic Leap One has not cables to link it to the power supply, only a cable to connect the headset to the lightpack, and the whole battery life is approximatively of 3 hours.
The controller follows the design of the other two components (All of these components are

beautifully designed with colorful lights) and has a number of buttons to allow the user to have a large choice of input commands.

Lightpack   As we said, the lightpack component is the elaboration unit of Magic Leap One. The central processor is a Nvidia Tegra X2 (Parker) SoC with two Denver 2.0 64-bit cores and four ARM Cortex A57 64-bit cores; the Graphics Adapter is an Integrated Pascal-based GPU with 256 CUDA cores; the RAM memory holds 8GB while the Hard-Disk 128 of which 95 available. Moreover it is equipped with Bluetooth and Wi-Fi systems, and a 3.5 mini-jack for your own headphones.

Controller   It is a Haptic device[7] with a built-in rechargeable battery (up to 7.5 hours continuous use) and some leds to make the illumination ring. The input is stored by means of a trackpad touch-sensitive and three buttons: Trigger Button, Digital Bumper Button and Digital Home Button. [11]

Cameras and Sensors   The device supports the 6 DoF tracking (section 3.2.2, fig. 3.6) of the headset, hands and controller, so you can move around the room without constraints and looking at the wires or base stations. In addition, the headset is darkened around the edges which slightly reduces the interaction with the real world and allows a 40° FoV, little bigger than HoloLens's one.
Magic Leap One exploits also the inside-out tracking principle (section 3.2.2.2, fig. 3.8).
Content creation begins in the lightpack. It provides power and handles the processing, sending image and sound data to the headset. Meanwhile, the lightwear headset tracks the controller's position and orientation, and maps your surroundings to help insert the virtual elements.

Optical Lenses   Like HoloLens, Magic Leap One bases on the optical see through principle enhanced by a new technology named "Virtual Retinal display" (VRD), which draws a raster display visualizing the holograms directly onto the retina of the eye. In the headset the technology of pupils tracking is used, so objects of virtual reality 'know' what exactly a user is looking at. [30]
The rounded lenses are set on the lightwear and can be personalized according to the preferences and necessities of the user. This is an advantage because you can play with the visor

[7] Haptic technology, also known as kinaesthetic communication or 3D touch, refers to any technology that can create an experience of touch by applying forces, vibrations, or motions to the user.

without your personal glasses, but if you want to share your visor with other people it results more complicated.

Spatial Sound In addition to a standard headphone output, it is necessary to mention the built-in speakers that are located on the inside of the mount. They are similar to HoloLens's ones and create the effect of finding the source of sound directly in the room.

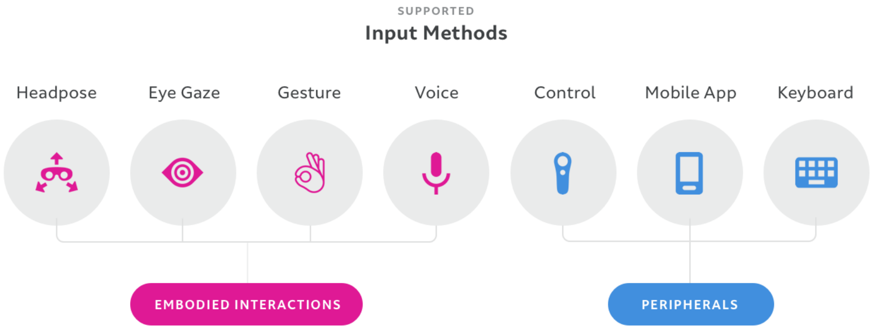


Figure 3.15: Magic Leap One's supported input methods [11]

3.4.2 Interface

The interaction with the environment occurs in different ways. The first, just mentioned is through the controller with its trackpad and buttons, which has its own tracking system and works very well. But also here there is a hand-tracking system which actually supports the 8 pre-defined gestures [fig. 3.16] and accurately tracks the placement of individual fingers.
Moreover Magic Leap One supports voice commands like HoloLens, and introduces in addition the innovative eye-tracking technology.
The supported input methods can be categorized as in figure 3.15.

3.4.3 Uses

The uses of Magic Leap are strongly similar to the HoloLens's ones. Also this device covers the whole VR spectrum and can represent all the environments, ranging from a self-stand Virtuality to a Mixed ambience, to an Augmentation of the real world (section 2.2).
Special care is needed by its operative system.

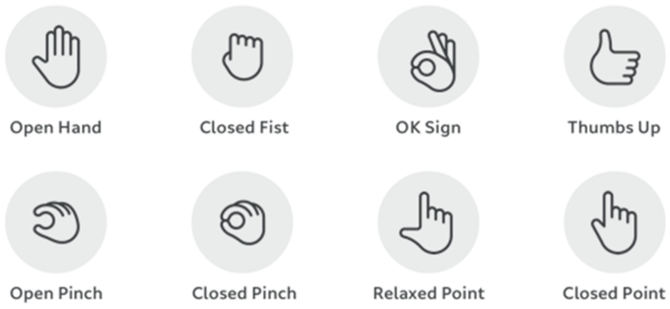


Figure 3.16: Magic Leap One's gesture mapping

> Lumin OS is fully optimized for environment recognition, persistent digital content and the performance to power high-fidelity visual experiences that turns your wildest imagination into even wilder realities. [11]

Magic Leap has published a development kit, like HoloLens, to create products for Lumin OS.
A quick comparison of the two presented devices can be resumed in table 3.1.
According to Azuma's belief[8], optical devices such as HoloLens and Magic Leap One are on the true path toward one day replacing nearly all other computing interfaces.

	HoloLens	Magic Leap One
Company	Microsoft	Magic Leap
Price	3000 $	2295 $
Platform	Microsoft	Lumin OS (Linux based)
Weight	579 g	345 g glasses, 415 g compute device
Battery Life	2.5 - 5.5 hours	3 hours average
Connectivity	Bluetooth, Wi-Fi	Bluetooth, Wi-Fi
Horizontal FoV	35^o	40^o
Display	Stereoscopic see-through	Stereoscopic see-through
Prescription inserts/glasses	Wear with personal glasses	Inserts available from third-party
AR tracking (6DoF)	Built-in	Built-in
User Interaction	Gesture, voice, gaze	6DoF controller, gesture, voice, eye tracking

Table 3.1: Hololens and Magic Leap One specifics' summary table

[8] http://www.imaging.org/Site/IST/Conferences/EI/Symposium_Overview.aspx?New_ContentCollectionOrganizerCommon=2

An idea that is not dangerous is unworthy
of being called an idea at all.

- Oscar Wilde -

4

Project: A Mixed Reality application in medical therapy

In this chapter the Unity 3D engine is introduced and the project developed is presented.

4.1 Introduction to Unity 3D

Unity 3D is a cross-platform game engine and a multi-platform integrated IDE, commonly used to create interactive visual content and video games. Its user-friendly interface, its networking ability and its power in generating physics, graphics and audio, all contribute to make it one of the most popular software used.
As anticipated in section 2.4.1, Unity can export projects in a generic XR to fit the desired target reality and platform. Unity's main components are described below. [12]

Assets An asset is a generic file that can be used into a project, like a font, an audio clip or a 3D object. It can be built inside Unity or with another program and then imported into Unity. Online there is an official Asset Store, where you can find and download lots of assets for free and not.

Game Objects The Game Object is the main element of Unity.

> You can think of a GameObject as an empty cooking pot, and components as different ingredients that make up the recipe of your game. [12]

When an asset is carried into a game scene it becomes a particular Game Object, containing different "Components", which describe its properties and behaviors, resumed into the "Inspector panel" [fig. 4.2]; in particular each game object has his "Transform" component, containing his pose (location and rotation – section 3.2.2) and scale.
Unity provides some simple Game Objects like spheres, cubes, cylinders and similar.

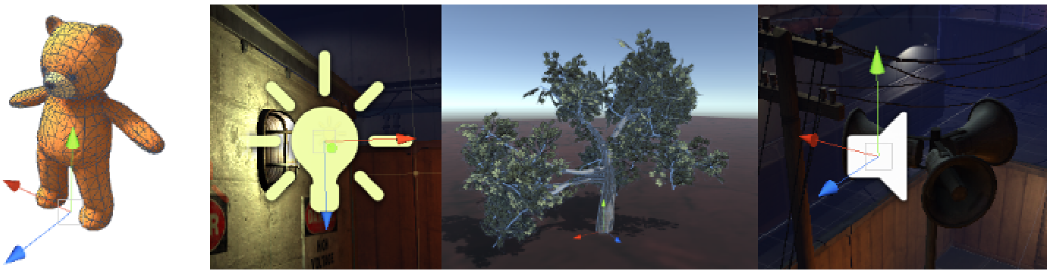


Figure 4.1: Four different types GameObjects; a character, a light, a tree and an audio source [12]

Components As anticipated, a Component is a property or behavior that can be assigned to a specific Game Object and can perform different functions. The most common Components are build and offered directly by Unity, for example the "Rigidbody Component" assigns to the object the necessary properties to respect the physical laws and the "Animator Component" imposes the conditions to play onto a Game Object an animation or another one.

Scripts It is also possible to attach a Script Component to an Object. A script allows to invoke some events, modify some properties and perform operations that are not included into the generic Components' functions.
The programming language supported are javascript and C#, actually used in the project.
The basic methods are "Start()" and "Update()" [listing 4.1]. The former includes the actions performed when the application is booted, while the latter is called every time the reference frame[1] refreshes.

[1] In physics, a frame of reference (or reference frame) consists of an abstract coordinate system and the set of physical reference points that uniquely fix (locate and orient) the coordinate system and standardize measurements

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Listing 4.1: Unity Script example

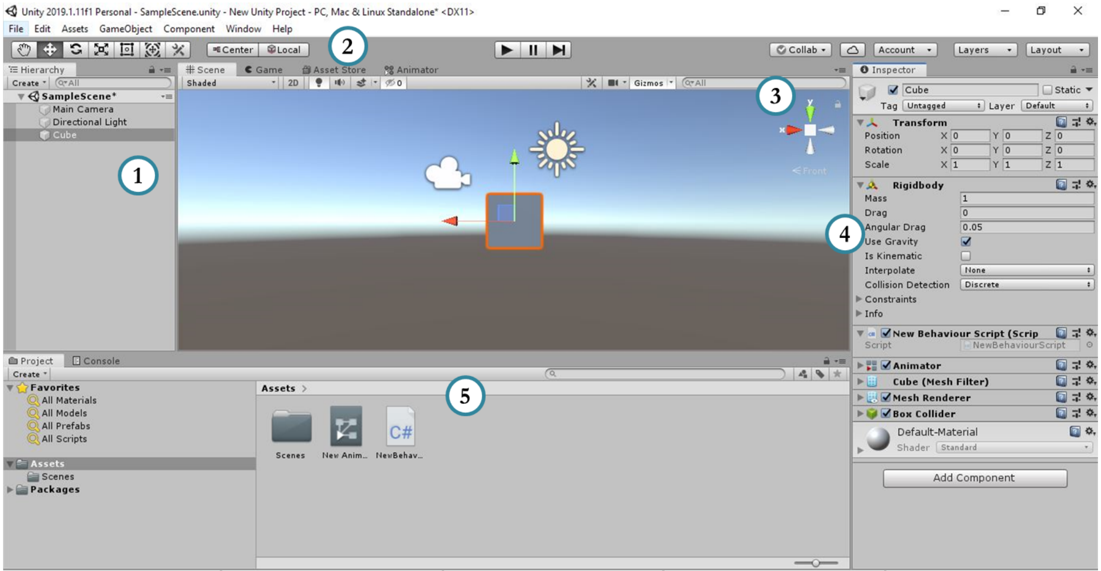
4.2 Gaming application in Medical therapy

We are now going to illustrate the project developed. It consists in a game, adapted and exported for three platforms: Android "Pie" (the target is API 28 but it fits also on previous versions) based handheld devices (section 3.2.1.2), HMDs HoloLens and Magic Leap One (sections 3.1.1, 3.3 and 3.4).

The finality of this game is the engagement of children while they are recovering in the pediatric units and need to undergo some medical intervention, or simply they are awaiting care, as we suggest in chapter 1.

As a matter of fact, since this applications need to carry out their task while surgeons/doctors are operating, some constraints need to be posed.

- The children cannot move too much or gesticulate, but they have to stay almost inactive. This represents a great limit for games, because the most of them using MR bases on gestures input and/or motion control systems.
- External operators, e.g. doctors, must have the possibility to place the virtual objects in the room conformingly with the medical operations that are taking place. It means that they have to be able to interact with MR world (section 2.2).


Unity 2019.1.11f1 Personal - SampleScene.unity - New Unity Project - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Window Help
Center Local
Collab Account Layers Layout
Hierarchy
Create
SampleScene*
Main Camera
Directional Light
Cube
Scene Game Asset Store Animator
Shaded 2D Gizmos
Front
Inspector
Project Console
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Scenes
Packages
Scenes New Anim... NewBehav...
Add Component
1
2
3
4
5

Cube Static
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ☑
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Info
New Behaviour Script (Scrip
Script NewBehaviourScript
Animator
Cube (Mesh Filter)
Mesh Renderer
Box Collider
Default-Material
Shader Standard

1. Hierarchy
The Hierarchy window contains a list of every GameObject in the current Scene.

2. Toolbar
It contains seven basic controls. You can use that to modify the position, rotation and size of the GameObjects

3. Scene
A Scene contains the environments and menus of your game.

4. Inspector
It displays detailed information about the currently selected GameObject, including all attached components and their properties.

5. Project panel
On the left it shows the structure of the project. On the right, the assets contained in the selected folder.

Figure 4.2: Unity 3D standard interface

Consequently the scene has to be stable and stationary once located, and not cover the whole 3D space around the user. The input has to be caught by means of voice commands (or at the limit by a stationary controller or touchpad).
Defined those boundaries, the games can be though in their complexity, paying attention to follow these guidelines.

4.3 MR Dance

MR dance is a simple game, developed with Unity 3D (section 4.1), where an Elvis look-alike character stands on a colorful dancefloor idling; he starts dancing and changes kind of dance according to some input commands given by the user.
The Elvis model and corresponding animations are open source and can be downloaded from `https://www.mixamo.com/`.
The nature of these commands is manifold. Each platform manages input commands differ-

ently, according to its input modality and devices, but this particular aspect will be deeply analyzed later; for now we explore the common features of our Unity project, stressing the properties of the Game Objects and Components, as well as the game logic and design.

Setting The scene (See figure 4.2) is single and contains three elements: a camera, a light and a Game Objects collection, which in turn groups three Game Objects: the Elvis character (whose name in the game is bigvegas), a dancefloor and a text object.
The camera is set with a "solidcolor" property, which allows to disable the background provided by Unity and to show the objects overlaid on real world when the project is deployed. The light is the one provided by default.

Game Logic The simple game logic provides that the kind of dance changes according to which key (a number from "0" to "4") is pressed. This events system is realized via script [listing 4.2, 4.3], and the script is attached to bigvegas's Game Object.
The script deals with animating bigvegas with the help of an Animator Component [fig. 4.4] also attached to bigvegas's Game Object and changes text into the text object and the associated music, to fit properly the chosen dance.
The Animator is realized by means of a complete graph which connects each node with one another and so permits the transition from any status to any other. Any time an input key is pressed, a parameter (an integer parameter called "Dancestyle") changes its value and induces the animator to make a transition.

In the Update method we have:

```
// Choose the desired dancestyle basing on the input key
if (Input.GetKey("1"))
{
    //Calls the Salsa Method
    Salsa();
}
```

Listing 4.2: Example of script associated to the dance change performed by means of a key pressed.

Where the method called is:

```
void Salsa()
{
    //Set the integer parameter to the value corresponding to the chosen dance
    anim.SetInteger("Dancestyle", 1);
    style.text = "Salsa!";

    //Stop the other sounds
    idleAudioSource.Stop();
    sambaAudioSource.Stop();
    hiphopAudioSource.Stop();
    bellyAudioSource.Stop();

    //and play the right audio
    salsaAudioSource.Play();
}
```

Listing 4.3: This script shows "Salsa" dancestyle method and relative actions performed.

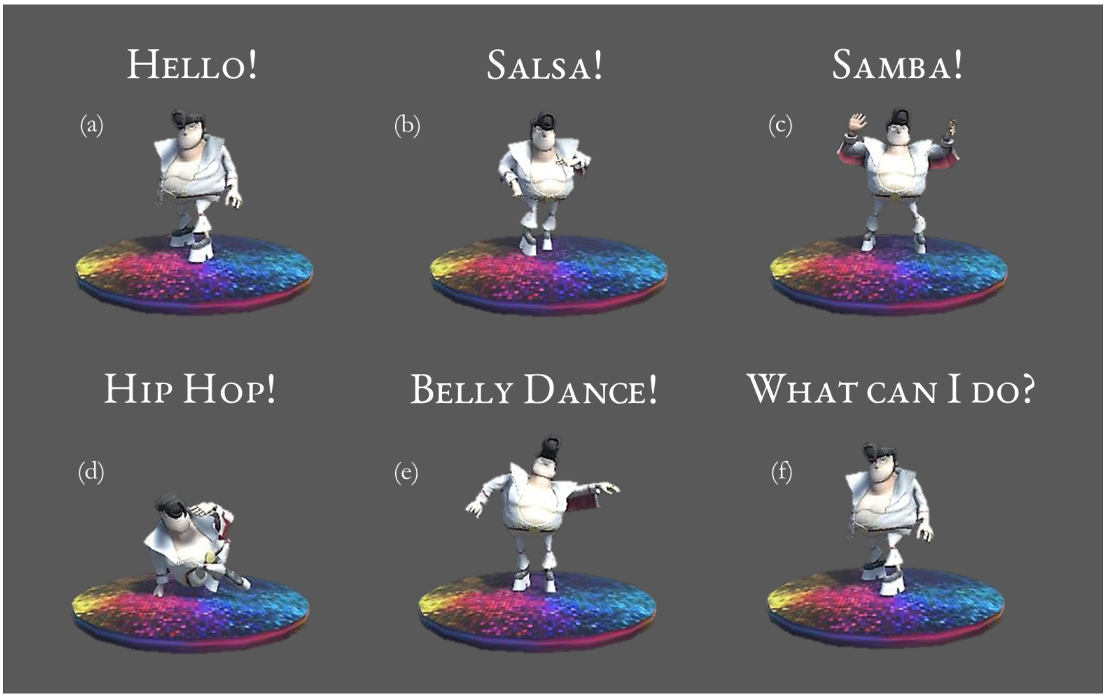


Figure 4.3: Different statuses of Elvis and text. (a) Initial status (b) Salsa (c) Samba (d) Hip Hop (e) Belly Dance (f) Stopped status. It's the same as initial, only text changes.

4.3.1 Android export

To build an application for a specific Android system, it is necessary to fit the parameters and specifications of that system. Let's start examining the user interface (UI) and the input

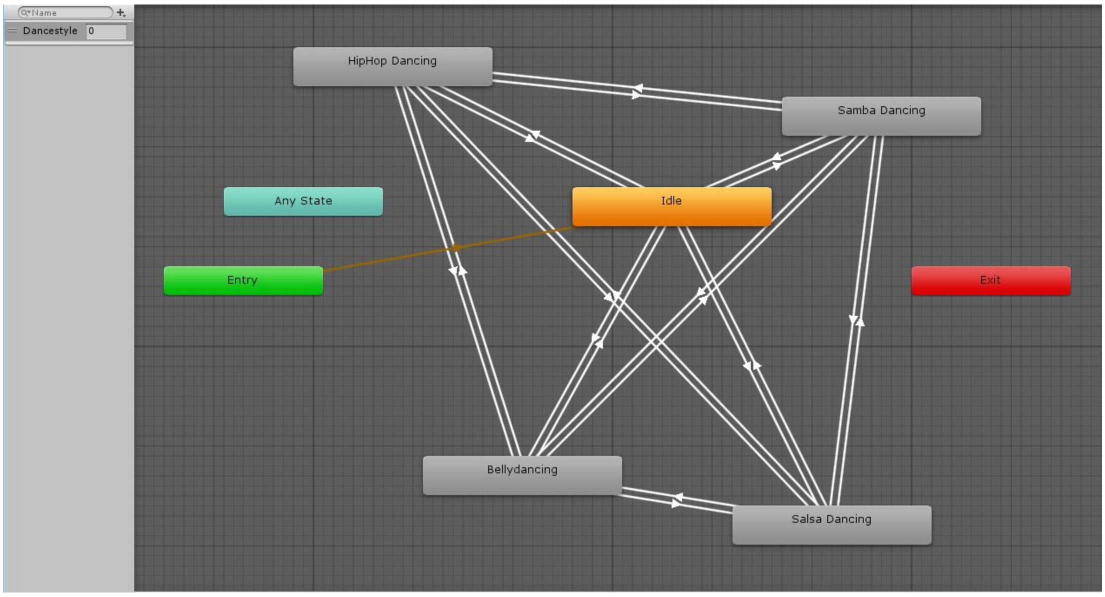


Figure 4.4: A complete graph representing the Elvis's animator. Each transition between statuses is possible, setting up a different value of the "Dancestyle' parameter.

modalities, explaining then how our script does change.

The Android based systems' main feature is the touchscreen, by means of which almost every input command is taken. Consequently it has not an input keyboard and the method described in section 4.3 can't work.
We need to include the touchpad, and therefore we place a "SingleStickInputControl", a Game Object from the "Standard Assets" (section 4.1) which draws a canvas on the screen, into the Hierarchy Panel (See figure 4.2). On the canvas we can situate UI elements, so we create five buttons to make bigvegas perform the five correspondent actions [fig. 4.5].
Since the input control was changed, we modify the script including a particular Unity Class (also imported from the "Standard Assets") named "CrossPlatformInput" which allows to perform actions as a consequence of a touch. So we assign a name to the buttons (e.g. Idle button's name is "Idle") and call them via script with the CrossPlatformInput method "GetButtonDown()" [listing 4.4].

In the Update method we have:

```
if (CrossPlatformInputManager.GetButtonDown("Idle"))
{
    Idle();
}
```

Listing 4.4: Example of script associated to the dance change by means of tochscreen buttons. Here "Idle" dancestyle is shown.

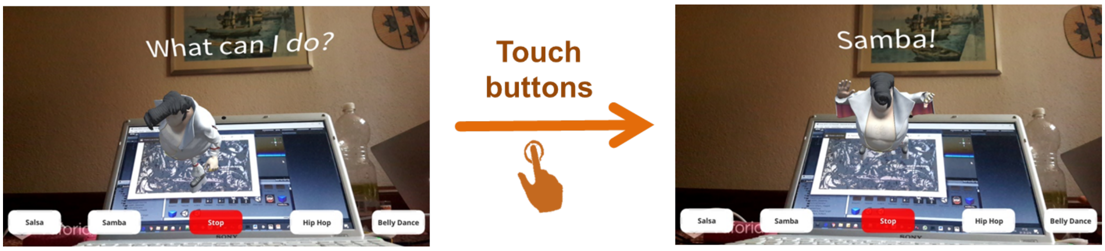


Figure 4.5: Elvis, who is staying on the marker, acts the same statuses as in figure 4.3, but the dance change happens when a button is pressed on the touchpad.

In order to place the objects correctly on the surrounding environment (section 2.2), we employed Vuforia[2], a Sofware Development Kit (SDK) that allows the marker based tracking (section 3.2.2.2). The chosen marker is highly contrasted and recognizable for the aim [fig. 4.6].

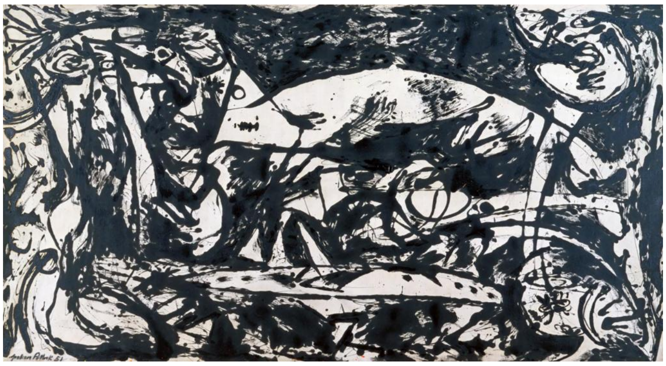


Figure 4.6: MRdance's marker.

4.3.2 HoloLens export

The HoloLens application is still different. Here the MR concept is achieved relying on the features offered by HoloLens: Spatial Mapping, Virtual Cursor, Gaze, Gestures and Voice

[2] https://www.vuforia.com

commands.

Particularly Voice commands (section 3.3.2) are used here, replacing keys, to change the dancestyle, via script as in listings 4.5, 4.6 and 4.7. The HoloLens library for Unity provides a “KeywordRecognizer” class and a “Dictionary” list, where you can store your own keywords. We use as keywords the names of the different dances, to make the system recognize them and consequently perform associated actions.
In addition we import from HoloLens’s website[3] the scripts relative to Gaze and Gestures.

Script is attached to the GameObjects collection. We import HoloLens’s libraries and define the variables:

```
//import the associated library
using UnityEngine.Windows.Speech;

//Define the Keyword Recognizer and the Dictionary
KeywordRecognizer keywordRecognizer = null;
Dictionary<string, System.Action> keywords = new Dictionary<string,
    System.Action>();
```

Listing 4.5: Definition of speech recognizer variables for HoloLens

Then we set the keywords:

```
keywords.Add("Samba", () =>
    {
        // Call the Samba method on every descendant object.
        this.BroadcastMessage("Samba");
    });
```

Listing 4.6: Example of setting the keyword “Samba”

And finally we add them to the dictionary:

```
// Tell the KeywordRecognizer about our keywords.
keywordRecognizer = new KeywordRecognizer(keywords.Keys.ToArray());

// Register a callback for the KeywordRecognizer and start recognizing
    !
keywordRecognizer.OnPhraseRecognized +=
    KeywordRecognizer_OnPhraseRecognized;
keywordRecognizer.Start();
```

Listing 4.7: Example of script adding the keywords to the dictionary

Instead of a marker, this application employs a markerless type of tracking (section 3.2.2.2), which bases on the “Spatial Mapping” [fig. 4.8].

[3] https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-101

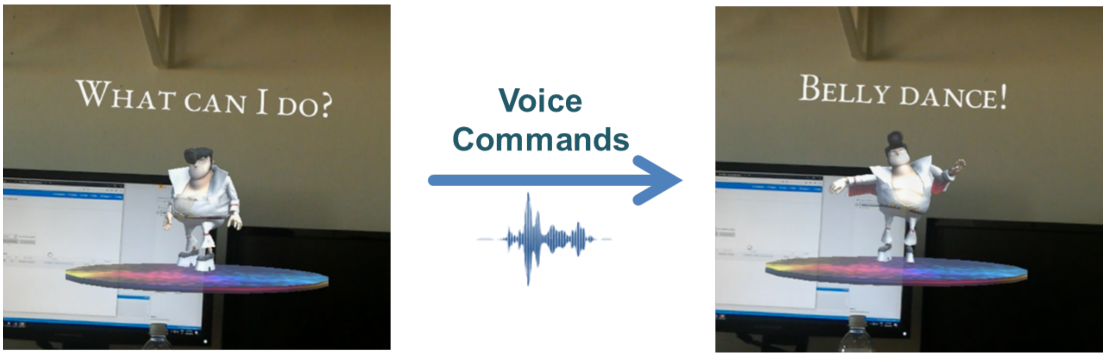


Figure 4.7: Elvis, located in the room, acts the same statuses as in figure 4.3, but the dance changes when the dance name is pronounced.

Spatial Mapping Spatial Mapping is the process of mapping real-world around the user into the virtual world. The mapping is divided into uniform chunks called Surfaces, oriented in the world in a way convenient to the system itself. It is important to highlight that transparent, black, and reflective surfaces do not work well with Spatial Mapping on the HoloLens. [12]

This mapping is necessary to place bigvegas where you want in the real space, similarly to what you can do with a marker, but using a "Cursor" to select and drag him in the desired position. Spatial Mapping and Cursor Game Objects are also imported from HoloLens's website.

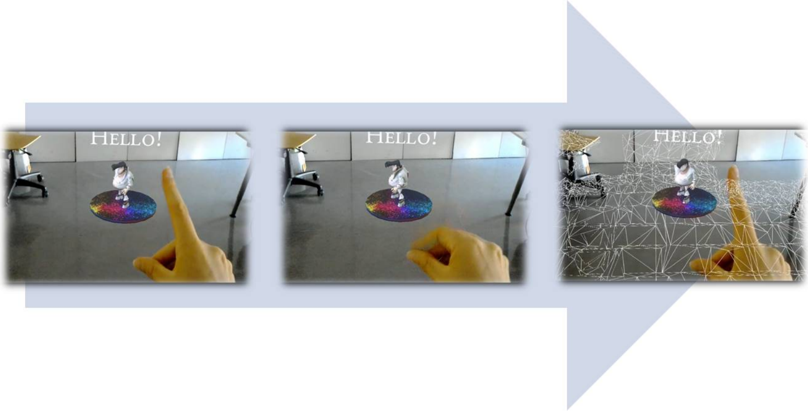


Figure 4.8: The "Air Tap" command on HoloLens allow us to place bigvegas in the real world.

4.3.3 Magic Leap export

The deployment on Magic Leap One requires different settings and a different input system. Here we use as input modality Magic Leap's integrated controller (section 3.4.2), instead of gestures and voice.

The controller is tracked with 6 degrees of freedom (section 3.2.2 and figure 3.6), thus we choose to take advantage of this unique feature to make bigvegas and its pool of Game Objects move around the real space according to the controller's movement. [fig. 4.9]

The dance change is here managed a little bit differently from the other platforms, because it changes dancestyle sequentially, advancing every time the Bumper Button, located on the controller, is pushed.

The whole control systems is handled via script (listing 4.8), exploiting Magic Leap's libraries and packages.

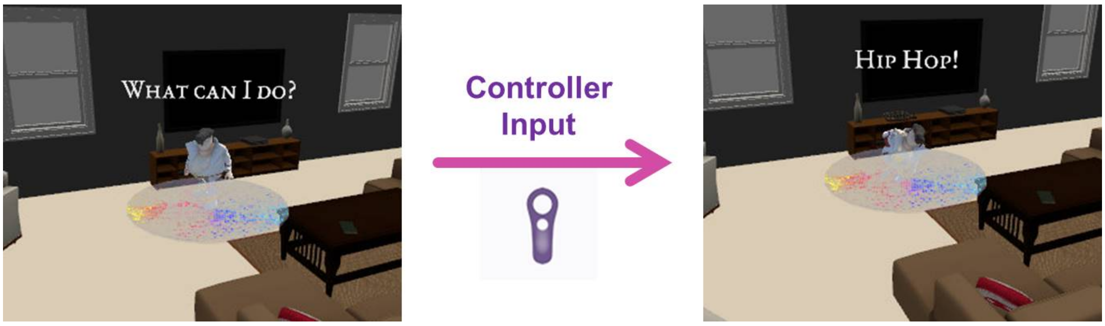


Figure 4.9: Elvis, located into a virtual room, acts the same statuses as in figure 4.3. He is moved and rotated according to the Controller's pose and the dance changes sequentially when the "Bumper" button is pressed.

In the Update method we have:

```
// Change Elvis's position according to the pose of the left
    controller with an offset
elvis.transform.position = controller.Position + Vector3(0,0,6);

// Check if the bumper button is down. In that case change dance
if (controller.IsBumperDown && !down)
{
    // Variable which stores the number of the dance
    dancenum = (dancenum + 1) % 5;

    // bumper is down. down variable is true
    down = true;

    switch (dancenum)
    {
        [...]
        // The fourth dance in order is Belly Dance
        case 4:
        {
            Belly();
            break;
        }
    }
}
// If bumper button is not down set the variable to false
if (!controller.IsBumperDown)
{
    down = false;
}
```

Listing 4.8: Example of script associated to the dance change by means of Magic Leap One's controller. It shows only "case 4", which corresponds to "Belly Dance".

5
Tests and Results

In this chapter the test performed on HoloLens and Magic Leap One and the results obtained were illustrated.

5.1 Evaluation setup

In order to make a comparison between HoloLens and Magic Leap One we performed a series of tests to evaluate some relevant objective and subjective features of the two HMDs.
Taken a pool of people, we trained them in the evaluation of the specific architectures. The group was made of seven university students, whose ages are included in range 23-28. Each user tested the MR dance app for about five minutes. The evaluation is performed by means of an integer scale from "1" to "10", where "1" stands for a very poor performance while "10" means that the MR experience was brilliant and it needs no improvements.
As for the application, they know nothing about and they discover what it consists of at the moment of their test, learning to interact with it.
Some people start their experience wearing Magic Leap One, while some other wearing HoloLens in order to have an overall unconditioned set of data.
In addition, we have also some experiences in which some scripts measure the objective performances, in particular the Frame Rate (expressed in Frames per Second (FPS)), which consists in how many updates the system is able to do every second.

5.1.1 Subjective Tests

Each person involved has to express a judgement on four subjective parameters:

- Image Quality, i.e., a subjective evaluation of the visual appearance of models;
- Flow of images, i.e., the smoothness of the visualization;
- Responsiveness of the controls, i.e., how quickly the device reacts to the users' commands;
- Usability.

We also asked each tester to leave a comment on their experience in general. The results obtained are stored in table 5.1.

Tests results

	Image Quality		Flow of images		Responsiveness		Usability	
	HoloLens	ML1	HoloLens	ML1	HoloLens	ML1	HoloLens	ML1
Person 1	7	8	7	8	6	9	8	5
Person 2	8	9	9	9	8	10	9	6
Person 3	7	9	9	9	7	10	8	6
Person 4	9	8	10	8	8	10	9	10
Person 5	8	6	9	10	8	9	8	6
Person 6	9	6	9	9	7	9	8	8
Person 7	8	4	8	8	7	9	4	5

Table 5.1: Results of each subjective test

After the experience, someone asserted that Magic Leap One is better than HoloLens as regards the quality of experience, while HoloLens provides good performance results granting a more natural experience, thanks to its better UI.

Someone other adds that both offer a nice Image Quality and Flow of images, with a little advantage for Magic Leap One. In their opinion despite Magic Leap One is more responsive, vocal commands and the absence of joystick make HoloLens the best choice in terms of Usability. Their thought is that the research has to go in that direction.

Someone stated that vocal commands are more practical than a joystick, while someone other disagree.

The judgements results not uniform as regards the Image Quality because Magic Leap One draws bright and maybe a little transparent Holograms, and some people dislike that feature while the others appreciate.

According to their judgements we can resume the evaluations in table 5.2 and we can assume that Magic Leap One is highly better in Responsiveness, while HoloLens prevails in Usability and Image Quality, and they are about at the same level as regards the Flow of images.

	HoloLens	Magic Leap One
Image Quality	8.000	7.143
Flow of images	8.714	8.714
Responsiveness	7.286	9.429
Usability	7.714	6.571

Table 5.2: Mean subjective values obtained.

5.1.2 Objective Tests

As regards objective measurements, we present in the table below (5.3) the results obtained from the same tests of section 5.1.1. Numbers report the average update rate (expressed in frames per second) which is strictly connected on the computational load of the device. This value, strongly depends on the interaction between the testing person and the virtual object (point-of-view, command given, etc.).

	HoloLens	Magic Leap One
Test 1	59.0367	58.9090
Test 2	59.0811	58.9357
Test 3	59.1429	58.8750
Test 4	59.0357	58.8214
Mean FPS	59.0741	58.8852

Table 5.3: Mean values of FPS obtained for each device.

Examining the table we can conclude that HoloLens's FPS is overall a little greater than that of Magic Leap One, and its performance under the same stress conditions is a little better, despite the worst subjective level of Responsiveness.
Anyway, from an objective measurement, we can conclude that the computational performance of these two devices is very similar.

5.2 Devices in comparison

Gathering the whole information from the tests and above discussions, we can draw the following conclusions.

Magic Leap One prevails in terms of FoV and Quality-of-Experience. It has also a better Responsiveness than HoloLens but it is lacking on the Usability level. Its HMD is really less heavy than HoloLens, but this is due to the fact that all the computation unit is in the lightpack, which is not comfortable to carry around; similar conclusions can be drawn for its control, which is really responsive, but not so different from many old joysticks.

HoloLens on the other hand is heavy to be worn for hours, but it is all integrated in the headset and the gestures and voice commands guarantee it a really user-friendly interface.

On the objective plan the two devices have about equal performances, according to the same value of the Flow of images. Talking about Image Quality, the results show that it is a really subjective parameter and neither between the two devices prevails.

Finally we must highlight that some people couldn't perform the test because they wear glasses and consequently they couldn't have a right experience with Magic Leap One.

6
Conclusions

The project developed in this thesis can be seen as a starting point for a major improvement and for other projects with a wider scope and complexity. For the moment it is a demonstrative work, which aims in being carried and presented into pediatric hospitals.
Thanks to the collaboration with Hannah Luxenberg, we aim at improving the current software in the functionalities as well as the models' characteristics. In the future, ethical and psychological issues will be considered as well.

In addition, this work allowed us to compare two of the most widely used headset for Mixed Reality: Microsoft's HoloLens and Magic Leap One. This will be an important starting point for future researches. More precisely, in our tests we will involve a wider spectrum of features and additional information to make these preliminary results more sound and robust. In particular:

- more heterogeneous people and devices will be involved in the experiments. In fact, working with only seven testers is a bit limiting and the evaluated numbers could turn out to be hardly replicable.

- the virtual objects visualized into the project of this thesis are not really complex from a computational point of view, since they are made of simple meshes with a limited number of vertices and polygons.
  The goal is to test the two visors using point clouds, which imply a higher compu-

tational stress on the device itself. These would challenge device's computational resources per each second, and therefore the objective measurement will be more precise.

- we will expand the objective metrics, going beyond the sole Frame Rate: this would allow a far better comparison of HoloLens and Magic Leap One platforms.

References

[1] D. Schmalstieg and T. Höllerer, Augmented Reality Principles and Practice, A. Weasley, Ed. Pearson, 2016.

[2] T. Strettona, T. Cochraneb, and V. Narayanb, "Exploring mobile mixed reality in healthcare higher education: A systematic review," Research in Learning Technology, 2018.

[3] https://medium.com/startux-net/the-differences-between-vr-ar-mr-27012ea1c5.

[4] B. Yoo, "Webizing mar contents - wxr library," University presentation, 2019.

[5] http://kiphaynes.com/wpress/?p=395.

[6] https://pupuru-blog.com/vr/oculus-quest-release.

[7] I. Hirotake, "Augmented reality: Fundamentals and nuclear related applications," Nuclear Safety and Simulation, Vol. 1, Number 1, 2010.

[8] B. Thomas and M. Billinghurst, "Ar technology: Tracking," University presentation, 2016.

[9] T. Dilani, "Markerless inspection of ar using fingertip tracking," Conference presentation, 2015.

[10] M. Valoriani, Mixed Reality: dalle demo a un prodotto, 2016.

[11] https://www.magicleap.com/magic-leap-one, "Magic leap official site."

[12] Unity3D Manual - Glossary.

[13] E. Costanza, A. Kunz, and M. Fjeld, "Mixed reality: A survey," 2014.

[14] G. Burdea and P. Coiffet, Virtual Reality Technology, 1994.

[15] R. T. Azuma, "A survey of augmented reality," Teleoperators and Virtual Environments 6, 1997.

[16] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," SPIE Vol. 2351, Telemanipulator and Telepresence Technologies, 1994.

[17] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," IEICE Transactions on Information Systems, Vol E77-D, No.12, 1994.

[18] J. Peddie, Augmented Reality Where We Will All Live. Springer, 2017.

[19] M. Weiser, "The computer for the 21st century," Scientific American Ubicomp Paper after Sci Am editing, 1991.

[20] G. Schall, "Mobile augmented reality for human scale interaction with geospatial models," 2011.

[21] S. Mann, "Code of ethics on human augmentation," 2014.

[22] S. Mann, T. Furness, Y. Yuan, J. Iorio, and Z. Wang, "All reality: Virtual, augmented and mixed (x), mediated (x,y), and multimediated reality," arXiv:1804.08386v1, 2018.

[23] J. Rousseau, "The laws of mixed reality — without the rose-colored glasses," 2016.

[24] Azuma, Baillot, Behringer, Feiner, Julier, and MacIntyre, "Recent advances in augmented reality," IEEE Computer Graphics Applications, 2001.

[25] S. Aukstakalnis, Practical Augmented Reality, 2017.

[26] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes, Computer Graphics - principles and practice, 1997.

[27] O. Grinchuk, V. Lebedev, and V. Lempitsky, "Learnable visual markers."

[28] A. Pinz, M. Brandner, H. Ganster, A. Kusej, P. Lang, and M. Ribo, "Hybrid tracking for augmented reality," 2002.

[29] S. Karthika, P. Praveena, and M.GokilaMani, "Hololens," International Journal of Computer Science and Mobile Computing, 2017.

[30] https://www.niora.net/en/p/magic_leap_one, "Niora," 2019.

Acknowledgments

Vorrei ringraziare tutte le persone che mi sono state accanto, che mi hanno aiutato e supportato nel mio percorso.

Per prima la mia famiglia, i miei genitori e mio fratello, i miei nonni e il mio Daniele, che mi vogliono bene e mi sono sempre stati vicini.

Poi vorrei ringraziare il professor Milani, Sebastiano, Fabio e i ragazzi del laboratorio LTTM, che mi hanno seguito, aiutato e dato molti utili suggerimenti per la tesi e per i miei studi futuri.

Ringrazio anche Hannah e Alberto Tono, TACTIC studio, Dreamship e Alumni Unipd "San Francisco and Bay Area Chapter", che mi hanno dato questa grande opportunità, di mettere a frutto le mie capacità e passioni e poter lavorare per uno scopo così interessante.

E infine ringrazio la mia professoressa di Inglese delle superiori che mi ha dato una base sufficientemente buona nella lingua per portare a termine questo lavoro.