

POLITECNICO DI TORINO

Master's Degree in Computer Engineering



Master's Degree Thesis

Augmented Reality to Support Motion Capture Shooting

Supervisors

Prof. Fabrizio LAMBERTI

Dr. Alberto CANNAVÒ

Dr. Filippo Gabriele PRATTICÒ

Candidate

Alberto BRUNO

July 2022

Abstract

Technological improvements have radically changed the way films involving VFX as well as video games are made. From green screen to ledwalls, from keyframe animation to motion capture, from paper storyboards to 3D visual storyboards, from normal cameras to simulcams and so on. Today, it is much easier to use technology to reduce costs and improve the quality of these products. On the other hand, all these changes have confronted actors with new scenarios, such as acting in the presence of green backgrounds or other actors wearing motion capture suits, which could be real challenges for them and add a greater workload in terms of imagination to understand what their character is actually seeing and feeling.

This thesis aims to demonstrate how augmented reality can be useful for actors when shooting scenes in which motion capture is involved. The idea is to have the actor use an OST-HMD (Optical See-Through Head-Mounted Display) that allows him to see, in real time, a 3D virtual avatar superimposed on the actor who is controlling him through the motion capture suit, together with the other real and virtual elements of the scene. In this way, the actor could become more familiar with the virtual character with whom he or she is interacting and with the environment in general, better understand what the character being played is facing and feeling, and thus, hopefully, play the scene better.

To this end, a framework was created that, using a visor of the indicated type (in particular, a Microsoft HoloLens device) and the Optitrack motion capture system, allows the user wearing the visor to see in real time an animated avatar superimposed on the actor traced by the optical system.

In order to show the usefulness of the proposed solution, an experiment was conducted involving a number of users. The experiment consisted in proposing the script of a scene to some volunteers, who had to prepare to act out the scene through rehearsals conducted either with a traditional approach (i.e., using only props) or with the proposed tool (i.e., using the augmented reality viewer). The acquired subjective and objective data were analysed and compared. The results obtained will allow further steps to be taken in this promising field of research.

Acknowledgements

“Give a man a fish and you feed him for a day. Teach a man to fish and you feed him for a lifetime... leave the dude alone and he’ll figure it out.”
Louis CK

Table of Contents

List of Tables	VII
List of Figures	VIII
Acronyms	XIV
1 Introduction	1
1.1 Augmented Reality	2
1.1.1 Introduction	2
1.1.2 How AR-HMDs work	3
1.1.3 Current and Future Technologies	9
1.2 Motion Capture	11
1.2.1 Introduction	11
1.2.2 Types of Motion Capture and Usages	12
1.3 Motivation	20
1.3.1 Introduction	20
1.3.2 The VFX pipeline nowadays	21
1.3.3 Making a motion capture shoot	26
1.3.4 Actors challenges with motion capture	29
2 State of the Art	35
2.1 Introduction	35
2.2 Related Works	35
3 System Architecture	46
3.1 Introduction	46
3.2 Basic Idea	46
3.3 The MotionHub Choice	47
3.4 Functioning	48

4	Technologies	51
4.1	Introduction	51
4.2	Optitrack	51
4.2.1	Introduction	51
4.2.2	Hardware	52
4.2.3	Software	55
4.3	MotionHub	57
4.4	HoloLens 1st Gen.	60
5	Implementation	64
5.1	Introduction	64
5.2	MotionHub Changes	65
5.2.1	Introduction	65
5.2.2	Receiving Single RigidBody Data	65
5.2.3	Communicating with Motive and Sending Signals	68
5.3	HoloLens Application	69
5.3.1	Introduction	69
5.3.2	Communication with MotionHub	69
5.3.3	Use of the Tracker’s Data	71
5.3.4	Application	76
6	Experiment	78
6.1	Introduction	78
6.2	References	78
6.2.1	First Reference	78
6.2.2	Second Reference	81
6.3	Use Cases	83
6.4	Experiment Procedure	86
6.5	Scene	87
6.5.1	Requirements	87
6.5.2	Script	88
6.6	Realization	95
6.6.1	Questionnaire	97
6.6.2	Pre-experience Questions	97
6.6.3	Post Three Times Rehearsal Questions	98
6.6.4	Post Shoot Questions	100
6.6.5	Post Both Rehearsal Methods questions	100
6.6.6	Augmented reality experience	101

7 Results	102
7.1 Introduction	102
7.2 Participants	102
7.3 Subjective Results	102
7.4 Objective Results	105
8 Conclusions	108
Bibliography	110

List of Tables

2.1	Mentioned works characteristics.	45
7.1	P-values regarding the subjective data.	105
7.2	P-values regarding the subjective data.	107

List of Figures

1.1	First HMD for AR/VR ever made [1].	2
1.2	Person wearing a Microsoft HoloLens device. [9]	4
1.3	A SLAM architecture scheme [11].	5
1.4	Example of a pixel analysis for a keypoint detection [12].	5
1.5	3D mesh reconstruction overlaid on real object [13].	6
1.6	2-D optical layout of an OST-HMD that use a free-form prism cemented with a free-form lens [14]	7
1.7	Some OST-HMD examples [15]	8
1.8	From left, HoloLens 2, Magic Leap and ThirdEye Gen X2.	9
1.9	FoV of current commercial HMD compared.	10
1.10	MojoLens prototype [17].	10
1.11	Andy Serkis during the shoot of The “Lord of The Rings: The Two Towers” (2002) posing in a motion capture suit to perform Gollum [18].	11
1.12	Types of motion capture systems [20].	12
1.13	Example of optical motion capture system made by OptiTrack [21].	13
1.14	Examples of active and passive markers [22].	14
1.15	Motion capture setup used in “Pirates of Caribbean” [23].	14
1.16	Circle markers identification [23].	15
1.17	On the left a scene from “Avengers” [24], on the right the fractal pattern used on the mocap suit [25]	15
1.18	On the left a scene from “Avengers: Infinity War” (2018) where the actor Josh Brolin tied a stick to his body to match his character height [26], on the right Will Smith playing the genie from “Aladdin” (2019) [27].	16
1.19	Mark Ruffalo playing Hulk in “Avengers: Endgame” (2019) using a motion capture suit with colored image markers [28].	16
1.20	Two examples of motion capture used in the making of of two videogames: on the left “Last Of Us” (2019) [29], on the right “Apex Legends” (2020) [30].	17

1.21	The use of motion capture for dogs in “Call of the Wild” (2020) [31].	17
1.22	An example of an inertial motion capturing made by Xsens using sensors to put on the body [32].	18
1.23	Example of the use of inertial mocap suit in cinema: Ariana Greenblatt and Sam Rockwell, who played the gorilla Ivan, on the set of “The one and only Ivan” (2020) [36].	19
1.24	VFX films pipeline presented by Andrew Whitehurst [38]	21
1.25	Differences between traditional and virtual production pipelines [39].	23
1.26	Example of simultaneous visualization on the set of “Welcome to Marwen” (2018) [41].	24
1.27	Two example of use of ledwalls: on the left a picture from the backstage of “The Mandalorian” (2019) [42], on the right a Sony Crystal-Led used on a photographic set [43].	25
1.28	Example of pipeline for a film that uses motion capture [44].	26
1.29	Actors standing in T-pose for the calibration of the mocap suit in the making of “The Witcher 3: Wild Hunt” (2015) [46].	28
1.30	Example of frame of a performance assembly provided by the editorial after a VFX shot production [45].	29
1.31	On the left two actress helped in acting in motion capture by a real time view of the virtual characters they were playing in the making of “Hellsblade: Senua’s sacrifice” (2017) [51], on the right an actor playing the role of a monkey in “Rise of the planet of apes” (2011) and the virtual result who could be seen live by the director [52]. . .	31
1.32	Terry notary playing the part of the dog Buck in the film “The Call of the Wild” (2020) [54].	32
1.33	Brian Cranston and Sam Rockwell on the set of the film “The One and Only Ivan” (2020) [34] while performing an emotional scene between a real character and a virtual character animated through motion capture.	33
1.34	Two examples of gimmicks used to help actors in the making of scenes in which motion capture is involved, trying to look alike the character they are playing in height/sized. On the left Josh Brolin playing Thanos [56], on the right Dan Stevens playing the beast [57]	33
2.1	Student lerning TaiChi using Immertai [58].	36
2.2	Woman wearing inertial motion capture sensors to use the virtual learning TaiChi system [59].	36
2.3	Using the AR HMD the user can see himself or herself playing with the golf-club and also the teacher avatar and his or her own avatar [61].	38

2.4	Through the app the user can see both virtual and real element in the scene [63].	39
2.5	Actor using AR to rehearse a scene of a figure with a sword [66]. . .	40
2.6	A pipeline that shows where the system presented by [65] can be useful in a film production.	40
2.7	Three steps of the experiment in [67] to prove that VR could be useful for acting rehearsal.	42
2.8	On the left two actors playing a motion capture scene, on the right the virtual scene in which they are immersed through their visors [68].	42
2.9	On top two actors playing a motion capture scene, in the middle the scene captured by the optical motion capture system, in the bottom the virtual 3D scene that they are seeing through their HMD [68]. .	44
3.1	The concept of system functioning.	47
3.2	Optitrack tracker placed on HoloLens to be recognized from the optical system	48
3.3	Pipeline of how the system works.	48
3.4	The MotionHub application receiving the skeleton and the visor data from Motive: on the right the visualization of the visor position and rotation data.	49
3.5	The Unity application receiving the skeleton and the visor data from motive and use it to move indirectly the main camera to change the view of the user.	50
4.1	On the left an example of Optitrack suit [70], on the right an example of an Optitrack motion capture setup with 16 cameras [71]	52
4.2	Optitrack camera [72].	52
4.3	The Optitrack camera's connection via Ethernet [73].	53
4.4	An Optitrack calibrator [74].	54
4.5	Two cameras watching the same 3D point [75].	55
4.6	The Motive interface.	55
4.7	Optitrack stream latency [76].	57
4.8	The unified skeleton structure streamed by MotionHub [69].	58
4.9	The MotionHub interface.	59
4.10	MotionHub workflow.	60
4.11	Microsoft HoloLens 1st Gen.	60
4.12	The headband with the wheel used to adjust the HoloLens to the user's head [78].	61
4.13	The HoloLens sensor bar [79].	62
4.14	The hand gesture recognized by the HoloLens used by the user to interact with the device.	62

4.15	The HoloLens lenses [80].	63
5.1	The path made by the tracked HoloLens position and rotation data from the OptiTrack system to the Unity application.	67
5.2	The MotionHub settings window before and after the interface modification for being able to connect with Motive from another machine.	68
5.3	The animate button added on the MotionHub interface to send animation signals.	69
5.4	Debug logs printed on the HoloLens every time a skeleton packet was received and processed.	70
5.5	How the two threads of the application works to give/take and use the data received from MotionHub.	71
5.6	How the HoloLens tracked data was used in the first approach. . . .	72
5.7	The AirTap gesture.	73
5.8	How the second approach works.	74
5.9	The second approach problem.	74
5.10	How the HoloLens tracked data is actually used to place the virtual camera in the correct position and rotation.	75
5.11	Menu.	76
5.12	The sliders in the calibration mode.	77
6.1	Actors shooting using VR for the experiment presented in [68]. . . .	79
6.2	The experiment done in [67].	81
6.3	On the left an illustration of the UC1, on the right an illustration of the UC2.	84
6.4	The experiment phases.	87
6.5	A frame from the scene “Friend like me” from the movie “Aladdin”(2019) [86].	89
6.6	Two scenarios (at the beginning, and at the end) of the proposed script.	93
6.7	The virtual elements present in the experiment script scene.	94
6.8	The volunteer’s view during the rehearsal with the traditional method (left), during the test with the AR method (centre) and during the shot (right).	95

6.9	From the left: 1) the tracked gloves used to track the tester hand, 2) the “magic book” and the knife prop used in the scene, 3) and 4) two props used to help the testers direct their gaze when interacting with the Genie (and Mini-Genie) during traditional rehearsal, and 5) a prop used to represent the dog and a laser pointer to give a reference on where to look to the tester when the dog was moving, during traditional rehearsal.	96
7.1	Features of the volunteers who participated in the experiment. . . .	103
7.2	SUS score results (the higher the better).	104
7.3	Questionnaire results (the higher the better).	104
7.4	An illustration of how the distance was calculated for objective metrics.	106
7.5	Data measured when shooting the scene after AR or traditional rehearsal (the lower the better).	107

Acronyms

Mocap

Motion Capture

AR

Augmented Reality

VR

Virtual Reality

VFX

Visual effects

MR

Mixed Reality

OST

Optical See-Through

HMD

Head-Mounted Display

FoV

Field of View

SRS

Spatial Reference System

BTS

Body Tracking System

SDK

Software Development Kit

OSC

Open Sound Control

IMU

Inertial Measurement Unit

DoF

Degrees of Freedom

SoC

System on Chip

CGI

Computer-generated imagery

Chapter 1

Introduction

Virtual Reality (VR) and Augmented Reality (AR) in the recent years are growing more and more. Big companies like Apple Microsoft, Google, Meta (formerly Facebook), etc. are investing a lot of money in these technologies. For example, Apple is about to release its first AR headset, Microsoft is developing its third MR glasses (HoloLens 3), Meta is pushing on the concept of *metaverse*, and so and so forth.

So far, indeed, AR is already being used for a number of different applications. For instance, to see directions on Google Maps over real streets, for home furnishing sales to see virtual 3D elements in our own house and better deciding whether to buy them or not, for creating remote assistants for work training, for medical training, for simulating surgery, etc.

This thesis work focuses on an application of AR in the field of cinematography and virtual productions, with the aim to support actors and directors.

For years, the world of video games and films, which involves visual effects(VFX) productions, has undergone various improvements through the use of increasingly better technologies which not only facilitated and enhanced the work of actors, directors and producers, but also helped to cut production costs. As a matter of example, while previously the directors had to wait days after the shooting to see how the film would turn out with the addition of special effects, now, since from the beginning of the production steps they have access to a 3D preview of the whole film; moreover, during the shootings they can already see in real time the special effects added in a good quality and modify them. This way, they can have more control over the film and its overall atmosphere.

Among the techniques used in films and video games for special effects is *motion capture*, often abbreviated *mocap*, an advanced animation technique used very frequently and in great demand, which facilitates the animators' task in making the digital characters hyper-realistic by guiding their movements through real actors who wears special suits. Acting in scenes which involves motion capture, however,

is not really easy for actors, as this technique can hide greater difficulties than normal acting, whether they act in the presence of other actors wearing motion capture suits or they are wearing the suits themselves.

This thesis explores this domain, by studying how AR can support the actors' job in the shooting of motion capture scenes. In particular, this chapter analyzes the concepts of AR and motion capture, and then focuses on the main questions behind the work: how can this technology be helpful to actors? What difficulties do they face when approaching scenes in which motion capture is involved?

1.1 Augmented Reality

1.1.1 Introduction

AR is a technology that interfaces virtual and physical spaces by superimposing virtual elements onto the real world. AR so can integrates all kinds of digital information (3D models, but also text, video) with the real environment, for example, with AR one can place virtual objects in a 3D real space, making the user think it is really there and it is occupying a real space in the world. Unlike what most people believe, AR is a pretty old technology; the first attempt of creating an head-mounted-display (HMD) – a display device, worn on the head or as part of a helmet that can let the user see a virtual 3D world or 3D elements in the real world – was by Ivan Sutherland at the University of Utah in 1968, so over 50 years ago (Fig. 1.1).

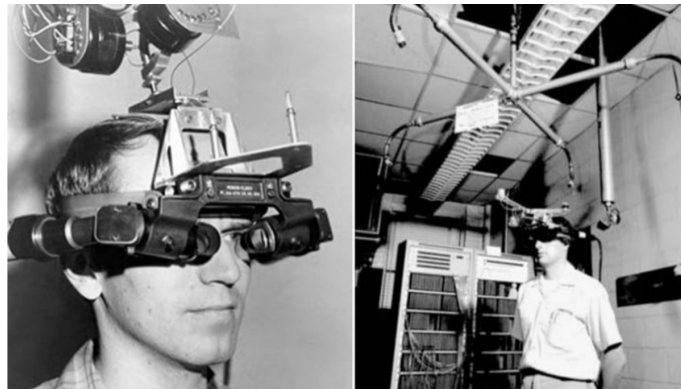


Figure 1.1: First HMD for AR/VR ever made [1].

The HMD was heavy and it had to be suspended from the ceiling, the system had a primitive interface, lacked realism, and the graphics comprising the virtual environment were simple wireframe rooms, but it was the first try of a revolutionary technology which people is still talking about today. Since then, AR has had up

and down in terms of popularity and interest of companies and universities, never becoming a mainstream technology.

Just in the recent few years the demand of AR solutions in a lot of various field has increased exponentially. Analyst predict that the AR market will reach \$198 billion in 2025 [2] which is a large portion of the economy within the next decade. This fast increasing of popularity is mainly caused by the improvements of technology in the recent years. In fact the main concerning about the AR is always been the usability, and the limits of technology ran against the real usability of an AR system: giant heavy HMD, small field of views (FoVs), bad understanding of the real world environment are just a few of the technology limits that had to be overcome to make AR an easy and valid mainstream technology.

However, as said, in the recent years big steps were made in this sense and AR is already a technology used in various kind of fields. Just think that many user already have in their pockets AR capable smartphones with sensors like gyroscope, GPS, accelerometer, and cameras that understand the space with algorithms or even LiDAR scanners like in the new Apple iPhone 13 Pro which detects the depth by means of a laser pulse, and use daily AR features even in mainstream apps like Instagram or Snapchat. Speaking about mobile phones, with Apple ARKit 5 and Android ARCore developers have a huge range of tools to use to build AR apps. And this toolkits are constantly updating, for example with ARKit 5 developers can place virtual objects with a specific latitude longitude and altitude collocating a virtual object in a specific real world position, or with ARCore now they can capture AR metadata from camera videos in real-time. All of this is awesome, and it is just about smartphones. Considering the HMDs, like the one showed in Fig. 1.2, which according to [3] are the most promising solutions within the AR market, one can say that they are already used in a wide spectrum of fields. AR has found application in medicine to assist specialists during pre-surgery, surgery and rehabilitation using HMDs and optical cameras [4], for surgical simulations [5], for making education more interesting and understandable [6], for retail [7], for remote assistance [8], for teleconferencing and hybrid meetings and so on. In short, AR is a technology that is increasingly becoming a part of everyone's life, and can really become the "next big thing".

1.1.2 How AR-HMDs work

AR, as said, consists in making the user see virtual 3D objects as if they are in the real environment. In order to do so special helmets have been invented and they are called HMDs. An HMD consists in one or two displays or projection technology integrated into an helmet or eyeglasses which, through the use of various sensors, specific algorithms and an high technology visualization system let the user see virtual elements in the 3D virtual world.



Figure 1.2: Person wearing a Microsoft HoloLens device. [9]

There are two main things that an AR system such as the HMD has to do in order to work properly.

1. Space understanding: make the HMD understand the 3D world around it and its position within that world in order to have a 3D space where to add the virtual elements;
2. Image visualization: make the user see the virtual elements over the real world in the correct way.

Space Understanding

Spacial understanding is a very complicated thing to achieve. It is essentially trying to make a computer understand the space around him as if it was a human being, understanding the depth, finding surfaces, creating a 3D representation of the world around it. Just think that, in order to understand the depth, humans uses more than 10 optical cues: stereopsis, occlusion, motion parallax, linear perspective, familiar size, relative size, texture gradient, shadows and so on [10]. So how does an AR device understand the depth and create a 3D model of the environment? The most used method is SLAM: simultaneous localization and mapping. SLAM is a class algorithms, which has succeeded other older algorithms like SIFT (Scale Invariant Feature Transform) or SURF (Speeded up robust features), for the construction and updating of a map of an unknown environment while simultaneously keeping track of the position of the device using the algorithm (in this case, the HMD). So it has two purposes: to build a map of an environment, and to locate the device within the environment. It is important to say that using a SLAM algorithm one does not get a definitive position of an element in space but a probability distribution of where it could be, and so other algorithms are used to calculate the positions based on uncertainties like MAP (Maximum A Posteriori) or BA (Bundle Adjustment).

A typical SLAM architecture is reported in [11] as it can be seen in Fig. 1.3. There are several elements taking part in this algorithm.

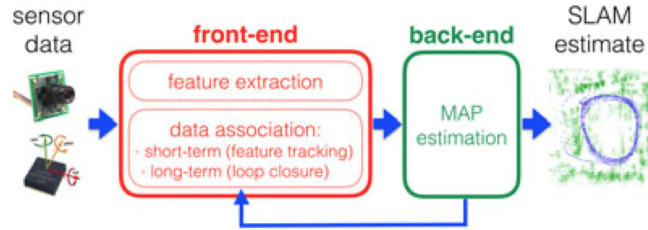


Figure 1.3: A SLAM architecture scheme [11].

First the data is captured by sensors which can include cameras but also accelerometers, gyroscopes, GPS sensors, depth sensors, etc. Secondly the feature extraction begins: 2D images are analyzed trying to find feature points. A feature point is a distinctive location in images; feature detection is a multi-step process and depends on the detection algorithm used. For example, an algorithm could perform a keypoint detection by analyzing the circular surroundings of each pixel in order to find a point which can easily be described by a collection of brightness data of his neighbour pixel. An example of this is shown in Fig. 1.4.

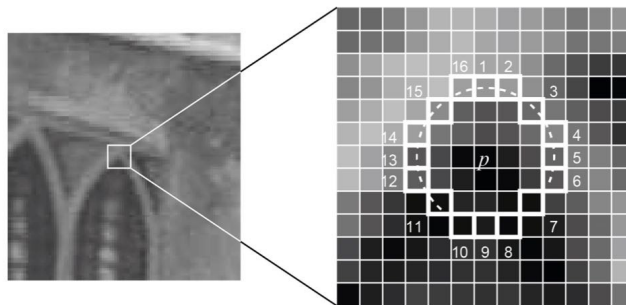


Figure 1.4: Example of a pixel analysis for a keypoint detection [12].

In other words, through this algorithm, the computer needs to find some points that are easily detectable in other frames too, and represent a real point in the physical world which can be located with a 3D position. This, integrated with the data from the other sensors, creates a series of keypoints with a 3D position found, called map points, for each frame. The back-end part takes care of establishing a relationship between different frames, it localizes the camera and moreover it handles the 3D reconstruction. It is important to say that the information of the environment is updated on each frame also considering the precedent frames captured, and also considering the movements made by the user (the camera

motion helps to perform the keypoint detection of the next frame). So the 3D reconstruction is more and more accurate as the user moves in the environment. So, substantially 3D points are created from a keypoints detection of the frame, then, this keypoints are triangulated with the keypoints from previous frames; this triangulation is made easier using the data from the movement of the user and also the data coming from the other sensor. Finally, the estimation is obtained: the final result containing the tracked features with their location and relations and the camera position in the environment. From this information 3D meshes can be created and overlaid to the environment. Typically triangles are used to create the mesh of the surface of the objects as it can be seen in Fig. 1.5.

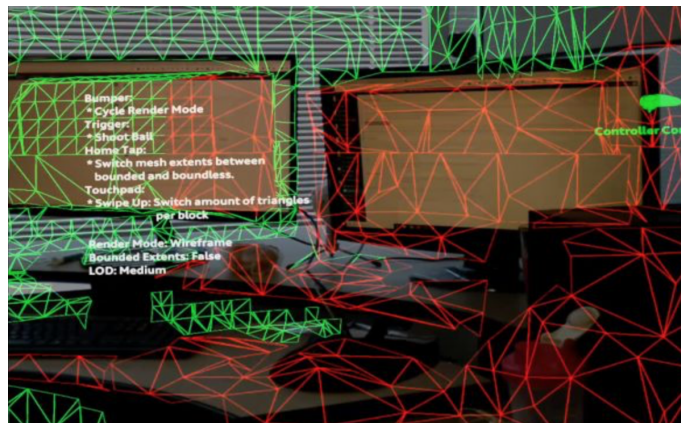


Figure 1.5: 3D mesh reconstruction overlaid to real object [13].

After this other algorithms can be used to transform the 3D triangle meshes in planes. This algorithms through the normal of the mesh can detect where is the floor or a roof or even detect a table. This can be very important also for performing a correct occlusion that happens when a virtual character is partially covered by a real object, for example a table, and so just a part of the virtual element needs to be seen from the user. However this are the basis of spatial understanding, there are plenty of algorithms and variations that depends specifically on the device, regarding which sensor it has, and what and how much data it has to understand the depth.

Image Visualization

Visualizing a digital 3D image to allow AR is another big technology challenge. Unlike the VR HMD in which OLC displays can be used right in front of the user's eyes to show the informations, AR, by definition, requires real world to be seen together with the virtual information, and for this reason, a see-through display is needed. But how can this be possible? How can a display be transparent

except for the part where the digital information is located? The first HMD to use what is now known as optical see-through technology, as already said previously on this chapter, was the one made by Ivan Sutherland in 1968. It is very important because it was the first that use the key idea to make seeing real things and artificial things together at the same time: use optical elements. This elements (in this case, half-silvered mirrors), were placed in the user's optical path to partially reflect a computer-generated image displayed on a miniature cathode-ray tube (CRT), and by doing this the user may still see the surroundings through the half-silvered mirror, but the displayed image is now superimposed on top of it. So the key for creating OST-HMD (optical-see-through HMD) is to use optical elements to project images over a lens, and doing so, leaving the rest of the view of reality unaltered. This can be seen in Fig. 1.6. This concept was the used in a lot of various attempt to create various types of OST-HMD, and, as the technology and the knowledge of optics grew, this HMD were more and more accurate.

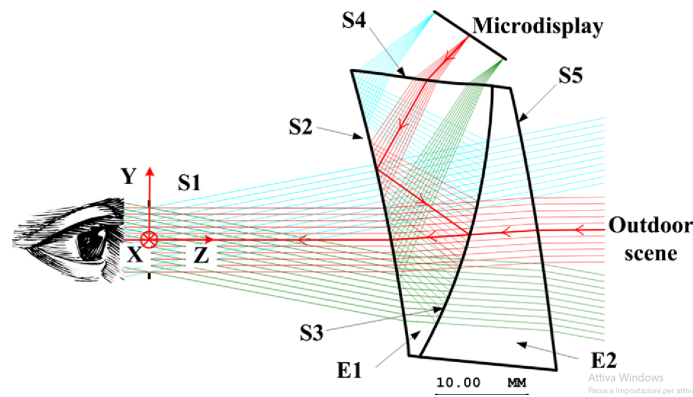


Figure 1.6: 2-D optical layout of an OST-HMD that use a free-form prism cemented with a free-form lens [14]

Speaking more specifically, to understand how a nowadays OST-HMD works, one needs to understand a few simple concept. The first is the difference between entrance pupil and exit pupil. The entrance pupil is the eye of the user that is seeing from behind the lenses, while the exit pupil is the projection that appears on the lenses, a virtual aperture in the optical system. Only light rays that pass through this virtual aperture can exit the system and be seen by the user, so the exit pupil is critical. The entrance pupil of the user's eye must match with the exit pupil so that the light rays enters in the entrance pupil in the right way. To do this is used a so called combiner. A combiner is an optical elements that, in various ways directs the light coming from the projector in the right direction. So the shoot out images must pass through a combiner that combines the projected image and the real world. The eye-box is another concept related to the exit pupil, it is the volume of space in which a viewable image can be created by an optical system.

When building OST-HMD is important to keep this eye-box as larger as possible, but to keep a large eye-box, a certain thickness of the optics is required, so solutions like waveguides are used to keep the eye-box as large as possible while maintaining a relatively small optics. The basic concept is to steer light towards the user's eyes by using internal reflection within the waveguide. Some HMD like HoloLens, uses a surface coating on the glass to create a series of defraction grating to enlarge the eye-box. [15] analyzed all the main technologies used in recent OST-HMD. Fig. 1.7 shows some OST-HMD examples.

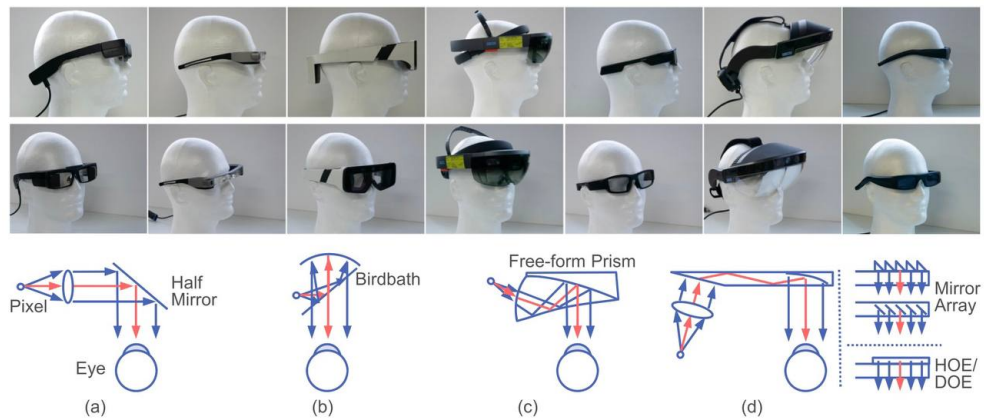


Figure 1.7: Some OST-HMD examples [15]

An half-mirror technology can be useful if one wants to keep the form of the lenses small and does not mind about the size of the eyebox. To achieve a large FoV, the Birdbath optics solution can be used; it combines a curved mirror lens that enlarge the FoV, but has issues with light efficiency and the design tends to be larger. The third solution is the one mentioned previously in this section, which is used also by the HoloLens: the free-form prism. This optical solution is good to achieve half-silvered mirror-like results while resolving the light problem letting light enter from various angles. It has been already said about the dimension problem of these type of optics and how waveguides can help it. Speaking more specifically the are different types of combiners that let waveguides to reflect the image correctly into the user's eye. The most simple ones are made by using a simple free-form mirror surface or a series of parallel mirrors or array of mirror layers, while a more optically advanced one is the holographic optical element (HOE) that can reflect the light at different angles, depending on the incoming light frequency. So this are the main technology solutions used to visualize 3D virtual elements when using OST-HMD.

1.1.3 Current and Future Technologies

Nowadays a lot of different AR technologies have been created, but what are the most AR technologies used? What could one expect in the future? At the moment it is possible to say that the best HMDs for AR are HoloLens 2 by Microsoft (2019), Magic Leap 1 by Magic Leap (2018) and X2 by Thirdeye Gen (2019)(Fig. 1.8).



Figure 1.8: From left, HoloLens 2, Magic Leap and ThirdEye Gen X2.

While HoloLens 2 and X2 are stand alone headset, Magic Leap keeps the computing separate and so and connected by a trailing cable, this means that a lighter headset with a pint-sized computer to put in the pocket is obtained. In [15], Itoh et al. analyze the main problems of OST-HMD's. They divided the problems into three main categories: the first concerns the difficulty of maintaining spatial realism, and this is related for example to the visual distrortions caused by imperfections in the display and optics; the second category is about maintaining temporal realism and concerns the problems of image update rates and general latency, in particular latency in AR which is involved in creating motion sickness as in VR but it can also create a misalignment between the virtual and the physical environment producing a visual incoherence; the third category is maintaining visual realism and includes issues like the realistic rendering of the AR scene, a coherent lightning, color reproduction, creating occlusion and depth accomodation. However the biggest technology limitation of OST-HMD is still the FoV, which is the main technical factor that is preventing a visually coherent presentation and the spread of AR HMDs. Currently the main commercials OST-HMD's FoV is 42° for the X2, wide 50° for the Magic Leap and 52° for the HoloLens 2 which are all good FoV considering that previous devices (like HoloLens 1) had just like 30° of FoV, but still not enough to have an optimal AR experience. Fig. 1.9 shows a comparison between OST-HMD FoV.

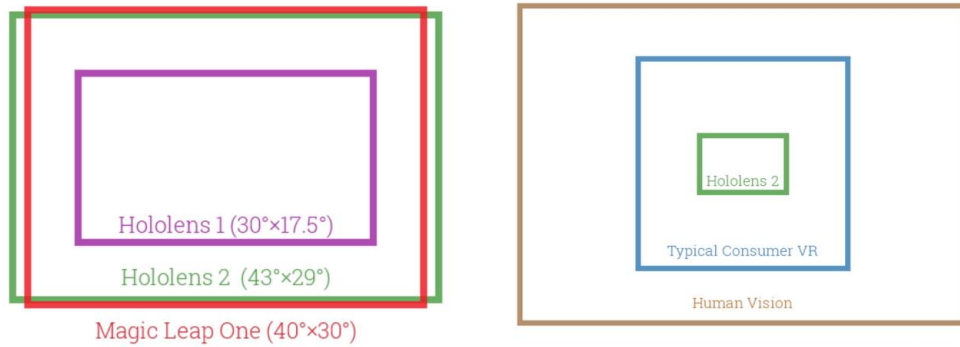


Figure 1.9: FoV of current commercial HMD compared.

Anyway, although these present technologies have these limitations the future for AR headset and AR in general seems bright. In fact Alex Kipman (the project manager of HoloLens), speaking to the Australian Financial Review, revealed that Microsoft is working on a new mixed reality (MR) headset (HoloLens 3) that will implement more powerful processor and an infinite FoV. Apple is also investing more and more in AR recently and there are a lot of rumors that they are about to launch their first AR visor. Furthermore Mojo Vision, a Californian company, is already developing a prototype for new AR eye contact lenses [16] (Fig. 1.10).

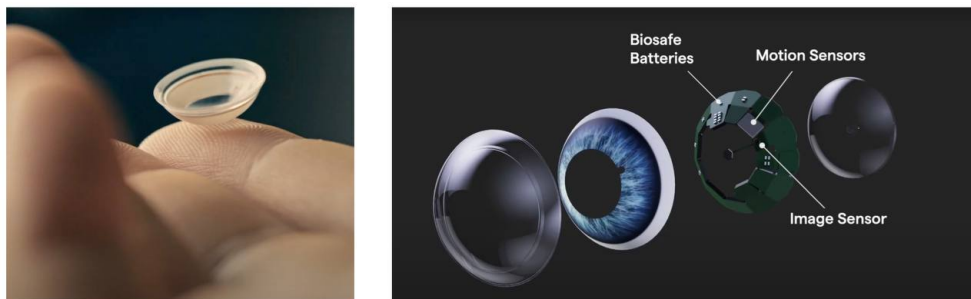


Figure 1.10: MojoLens prototype [17].

These are just rumors and prototypes but they attest to us that AR is a technology on the rise and it could be used more and more in the future. In this prospective, this work could be more and more relevant in the future when, with better AR headset with unlimited FoV and AR contact lenses, it would be so much better and easier to use for actors and directors in cinema and video games industries, overcoming a lot of problems that can occur and that will be analyzed more specifically in the use cases section.

1.2 Motion Capture

1.2.1 Introduction

Motion capture (or performance capture), is becoming a must in nowadays animated films and also in the video games industry. This technology has evolved drastically in past decades and has reach levels that was unimaginable to reach only 10/20 years ago, widely increasing the speed of animation, decreasing the work load on the animators, and creating a new way of perform for actors. The main idea behind mocap is to capture in some way the movements of a real world object (in this case the actor), converting it into data, and use that data to pilot virtual tridimensional objects (in this case virtual characters). This, as said, reduces the works of animation, speed up the entire process, and can give incredible realistic results.



Figure 1.11: Andy Serkis during the shoot of The “Lord of The Rings: The Two Towers” (2002) posing in a motion capture suit to perform Gollum [18].

There are a lot of examples in films about the use of mocap like the famous character Gollum in “The lord of the rings” (2001)(Fig. 1.11) or the apes in “Rise of the planet of apes” (2011) or na’vis in “Avatar” (2009) or more recent films like “Big” (2016), “War for the Planet of the Apes” (2017), “Aladdin (2019) or Marvel’s films like “Avengers: End Game” (2019) and so on. Mocap is also used in the process of making video games to perform the video scene or even the in-game movement like rest position, walking, or characters reactions to events in game. Examples of video games whose character animations was made with mocap are “Uncharted 4” (2016), “God of war” (2018), “Apex legends” (2019), “Destiny’s 2” (2020), “Friday the 13: the game” (2017), “The last of us part II” (2020), etc.

1.2.2 Types of Motion Capture and Usages

Speaking more technically there are many ways to capture the 3D data needed to animate a virtual character through mocap. In [19] S. Sharma et al. explore all the mocap systems for 3D animation. They categorize the mocap systems in marked-based and marker less. Marked based systems requires the actor to wear a suit or some sensors and can be acoustical, mechanical, magnetically or optical, while markerless systems do not require the performers to wear anything.

There are four types of mocap: mechanical, inertial, magnetic and optical as it can be seen in Fig. 1.12.

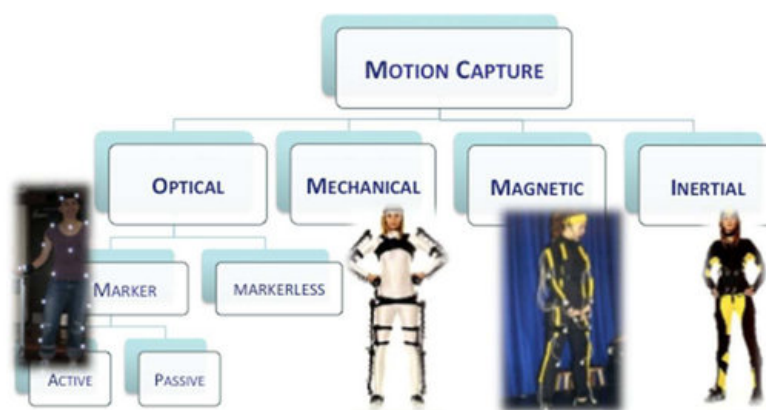


Figure 1.12: Types of motion capture systems [20].

In mechanical mocap, a performer attaches to his body a skeletal-like structure, and so the data are taken directly from this instrument. This method is obviously relatively uncomfortable for the performer but it is not affected by occlusions problems. Inertial mocap technology is based on miniature inertial sensors, bio mechanical models and sensor fusion algorithms. The motion data of the inertial sensors is often transmitted wirelessly to a computer, where the motion is recorded or viewed. However, this technology will be discussed more in detail later. Magnetic mocap systems on the contrary involves a transmitter who creates magnetic signal and some receivers positioned on the body of the performer. These systems calculate position and orientation by the relative magnetic flux of three orthogonal coils on both the transmitter and each receiver. This type of mocap is much cheaper than the optical one, can be used for real time applications, but is sensitive to magnetic interference and also working with this magnetic components makes this system really hard to be used in environments not designed for their use like the cinema and video games industries. In fact, in these industries the most common type of mocap system used, nowadays, is the optical one. This is due to the high accuracy of the data it generates, the fact that it is possible to track more than one

character at the same time and also that the tracking space in which actor moves could be very large and this gives movement freedom to actors and directors, and so it could be used in various type of scenarios. An example of optical system is the Vicon system, widely used in many film productions. But how does the optical system works? It consist in a series of cameras placed all around a large empty room as it can be seen in Fig. 1.13, calibrated to calculate the exact 3D position of markers that, in this case, are placed all over the actor's body. So the performer movements are registered by cameras from all the angles and the 3D data of all the track points placed on the actor are captured. Then this data is used to animate a virtual avatar on the computer.

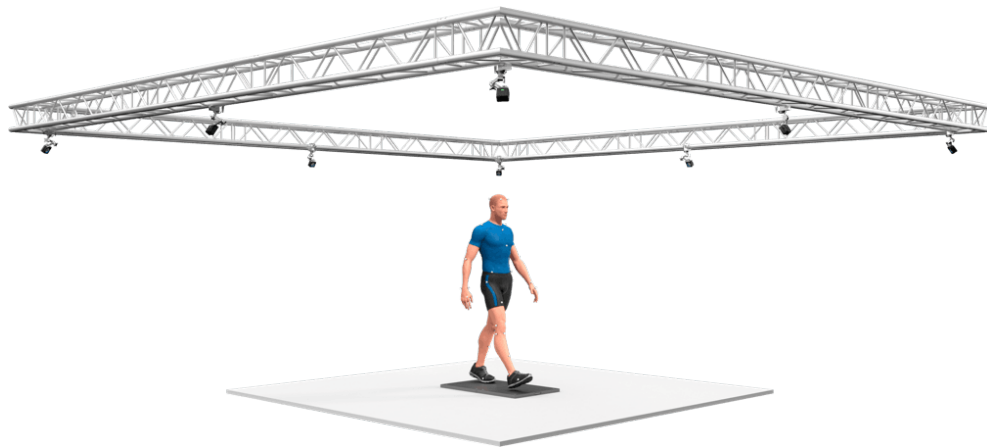


Figure 1.13: Example of optical motion capture system made by OptiTrack [21].

The optical mocap systems uses from two to 48 cameras to detect markers placed on actors. The dimension and shape of markers depends on the cameras resolutions. For each marker this system produces data with three degrees of freedom while the rotation information is taken from the relative orientation of three or more markers. An optical motion capture system can use active markers or passive markers. Passive markers are made with some spheres covered with a retro-reflective material, this markers are tracked by infrared cameras and it is usually used in acting mocap performances because it gives more accurate results. Active markers instead are LEDs that sends light signals to cameras by illuminating one LED at a time very quickly or multiple LEDs using a software to identify them by their relative positions. An example of the differences between these two types of markers is showed in Fig. 1.14.



Figure 1.14: Examples of active and passive markers [22].

However in film productions, often, some image markers are added. This usually happens when filming in situations in which not too many tracking cameras can be placed, like when shooting with mocap suited-actor in real context. The first film which implemented additional image markers over the optical mocap suits was for making the Davy Jones crew in “Pirates of Caribbean” (Fig. 1.15). The identification of the circle markers is showed in Fig. 1.16. They choose to use this technology because, having to shoot the motion capture scenes in various scenarios (and not in a studio with a lot of optical cameras), they needed a way to capture in the most accurate way possible all the characters position in various lighting conditions and with just few cameras because, by using rigid objects of known size and pattern that could be modeled in software, it is not necessary for each marker to be visible from multiple camera views.

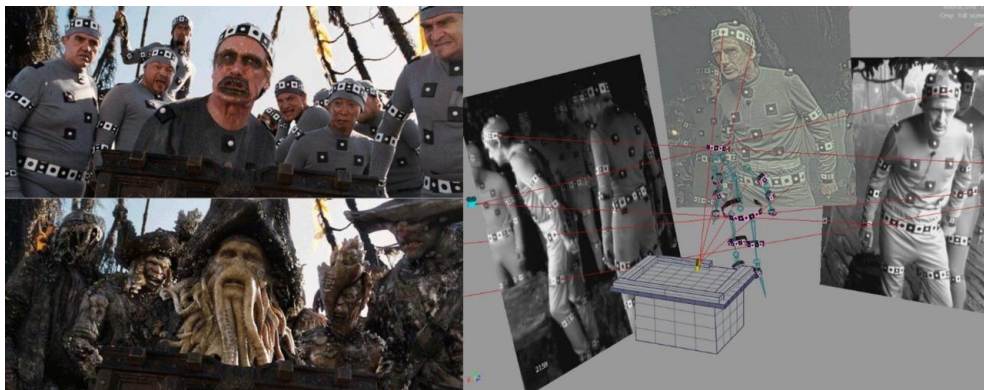


Figure 1.15: Motion capture setup used in “Pirates of Caribbean” [23].

John Knoll, the visual effect supervisor of “Pirates of Caribbean” states that

“unlike tracking markers with an optical mocap system, which requires a marker to be visible from multiple cameras in order to triangulate its position, if enough points around a band are visible, even from only a single view, one can solve its position and orientation in 3D space”.



Figure 1.16: Circle markers identification [23].

However this technology evolved in years. Since the “Avengers” (2012), for this type of scenes mocap suits with fractal pattern printed on it have begun to be used. In this case in particular Sierpinski triangle (which are a fractal pattern) are printed on the suit as can be seen in Fig. 1.17.



Figure 1.17: On the left a scene from “Avengers” [24], on the right the fractal pattern used on the mocap suit [25]

The idea with the fractal pattern is that now the tracker could automatically find features on the suit, no matter the focus or scale of the pattern in the footage. In fact fractals are by definition scale independent. So a fractal pattern is useful for motion capture because whether near or far there are always some visible points to track and its is also good to track in situation when the image is blurred, like in when the actor is in motion. And moreover they use triangle because they have identifiable points that can be tracked by computer vision algorithms, and these trackable points make up the polygon meshes used in 3D graphics. In the most recent films like “Avengers: Infinity War” (2018) or the live action film “Aladdin”

(2019), suits with fractal patterns where useful for virtual characters like Thanos the Genie (Fig. 1.18).

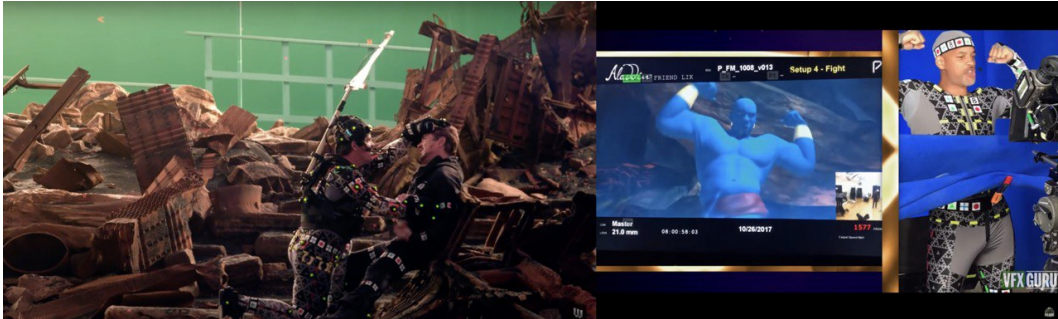


Figure 1.18: On the left a scene from “Avengers: Infinity War” (2018) where the actor Josh Brolin tied a stick to his body to match his character height [26], on the right Will Smith playing the genie from “Aladdin” (2019) [27].

Notice that in this figure the face of Thanos was put over the head of the actor to make Robert Downey Junior watch in the correct direction when staring at him. It has to be noticed that also the image markers have changes since the first “Pirates of Caribbean”. Now they are not just black/white circles in white/black squares but are made by various shape with various colors. An example of this is showed in Fig. 1.19 where is reported a moment of the making of “Avengers: Endgame”.



Figure 1.19: Mark Ruffalo playing Hulk in “Avengers: Endgame” (2019) using a motion capture suit with colored image markers [28].

Anyway in situation in which the set is not real (all actors are in a room made expressly for motion capture) “normal” optical suit are used. In this case spherical

markers are enough and no additional image markers are needed because a lot of cameras are used and so there are no problems with occlusions. This is used when characters played are totally in virtual world or when one needs just to capture a movement that will be used later on Computer-generated imagery (CGI) characters. This is the case of video games, in fact in this productions motion capture is largely used, just to mention a few recent video games in which has been used motion capture: “Star Wars Clone Wars (2020)”, “Titan Fall 2” (2016), apex legends (2019), “Destiny’s 2” (2020), “Friday the 13: The game” (2017), “The Dark Pictures: Little Hope” (2020) and so on. Two examples of video-games making of are showed in Fig. 1.20.



Figure 1.20: Two examples of motion capture used in the making of two videogames: on the left “Last Of Us” (2019) [29], on the right “Apex Legends” (2020) [30].

One last mentionable example of how motion capture is been used in the cinema and videogames industry is the film “Call of the Wild” (2020) in which, because the film required a dog made in CGI, they made a library of dog movements by using motion capture suits over dogs in a mocap studio (Fig. 1.21).

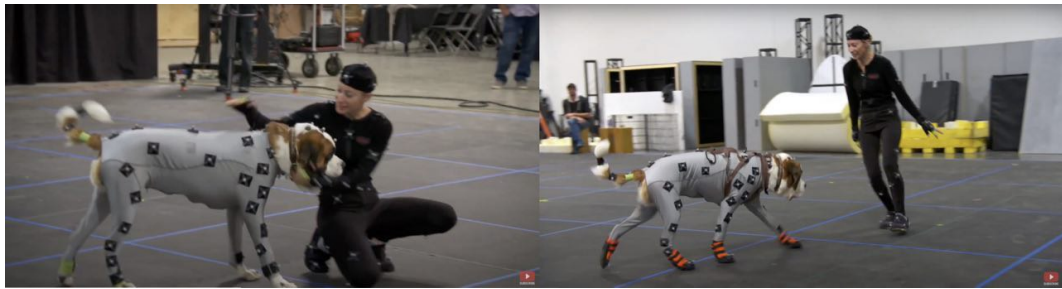


Figure 1.21: The use of motion capture for dogs in “Call of the Wild” (2020) [31].

Anyway, it must be said that optical systems have some disadvantages. Apart from the problems caused by occlusions and lightnings which can be easily overcome, the main problems is the use of the cameras which costs a lot (about 8000\$ every

six cameras), and the need of a lot of cameras to do a complete accurate tracking. This can be a problem for small productions of films or video games who needs to use motion capture to animate their character, considering also the fact that motion capture is important to reduce the costs of animating too. Luckily, technology has grown so fast that a lot of cheaper inertial type of motion capture systems has been created. New inertial systems like Noiton, Nansense, Xsens, Rokoko and others are now available for 5000/6000\$ on average. Unlike optical systems, this systems do not suffer for occlusion or lightning problems so it can be used regardless of the environment, they work with an accelerometer (for inclination) a magnetometer (for heading) and a gyroscope placed on every sensors on the body of the performer. Usually they also include a software system that improves the data read by the sensor to make better animations. Fig. 1.22 shows how the movement captured by the inertial system are visualized in the related software.

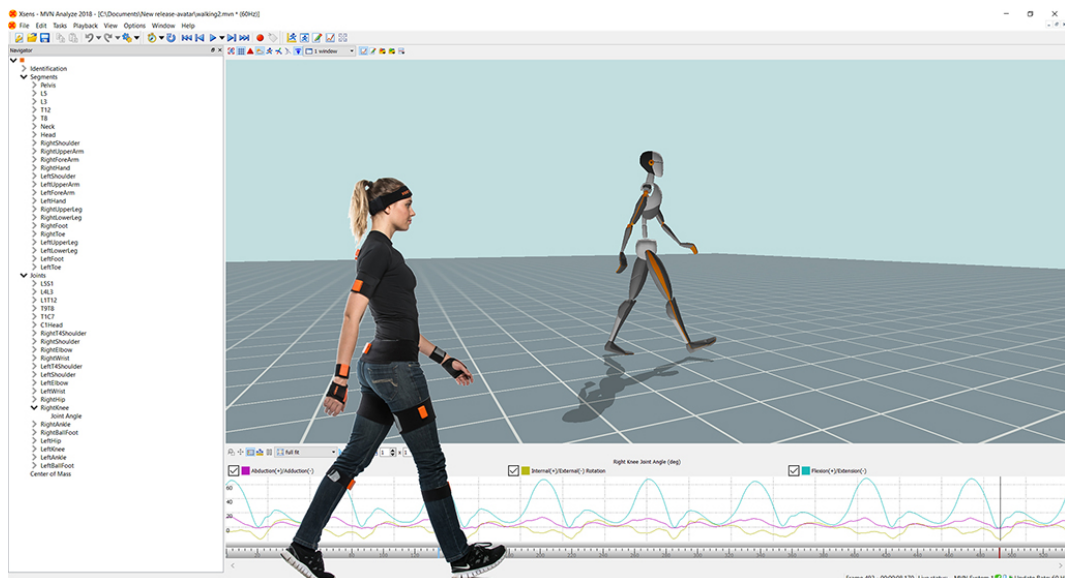


Figure 1.22: An example of an inertial motion capturing made by Xsens using sensors to put on the body [32].

Xsens presented their two new motion capture systems [33] (a suit and a set of sensor) that shows significant improvements to this technology. They affirm that there are major problems concerning inertial motion capturing due to the limits of accuracy that can be obtained using gyroscope, accelerometers and magnetometers alone and also by the magnetic distortion that may interfere. After acquiring data, Xsens system uses a biomechanical model to create a more accurate and fluid animation. However the old biomechanical model has limited accuracy when applied to a wide range of subjects. So, considering all these problems, they work

very hard to improve this technologies and overcome them creating a new engine with an advanced biomechanical model and immune to the effects of magnetic distortion. In [33] they compare their new motion capture system with an optical mocap system (eight-cameras Qualisys system) and the results are amazing. The differences between their inertial mocap system and the optical system were minimal, and Xsens system was able to track consistently human body kinematics in any environment. In this context, it should be mentioned the film “The one and only Ivan” (2020) in which they used an Xsens inertial suit adding markers on it. So the mocap system obtained was an hybrid between optical and inertial motion capture [34] (Fig. 1.23). It is good to use hybrid systems because combining inertial sensors and optical sensors reduce occlusions and improve the ability to track without having to manually clean up data. An Xsens suit has also been used in the making of the film Ted [35] for animate the main character Ted, a teddy bear.

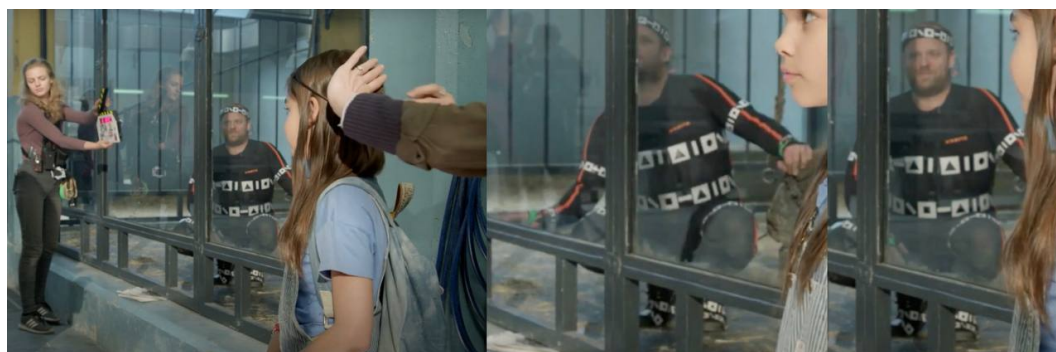


Figure 1.23: Example of the use of inertial mocap suit in cinema: Ariana Greenblatt and Sam Rockwell, who played the gorilla Ivan, on the set of “The one and only Ivan” (2020) [36].

Anyhow, although optical motion capture systems and, in some cases, inertial systems are the mostly used nowadays in film and videogames production, considering all the progress made by machine learning technologies and artificial intelligence used for image and video processing, which are becoming more and more accurate in tracking moments, the future of this optical technologies will likely involve optical markerless motion capture systems. This would give more freedom to actors and reduce the overall costs due to the lack to suits in this systems, increasing the cost of the software for the image processing. In the recent Vicon x Thiao Markerless webinar [37] in 2021, they presented an optical markerless system for tracking human body movements, that uses Vicon cameras and Thiao softwares that, by a neural network, recognize humans and, using multi body 3D pose estimations create the 3D skeleton of the person and finally have the validation of the 3D model created with a trained dataset. In conclusion, motion capture is a technology that

has been used, is used and will continue to be used in the production of films and videogames, and which will certainly continue to evolve technologically, increasingly meeting the needs of actors and directors.

1.3 Motivation

1.3.1 Introduction

The technological evolutions of recent years have taken hold in many fields including film production, and have concerned in particular films where virtual effects (VFX) are present. Using technology in VFX cinema offers mainly three benefits: the first is to improve films and make it possible to create scenes that were previously impossible, the second is to give the director and the actors more tools to be aware of what they are going to shoot (through for example ledwalls or previews which will be discussed later) and the third is to decrease the workload and therefore the production costs. In recent years, the increasing use of technology in film production has changed the process itself that leads to the creation of a film in which there are special effects: from a traditional way of producing films to something new that adapts and tries to make the most of the benefits given by technology and which has been defined as “virtual production”. All these changes therefore make the film industry something in constant change and evolution and directors, actors and all the other people involved in the production of films have had to adapt and are continuing to adapt to the use of new technologies and new production paradigms in film productions.

Among the many technological innovations that have changed the world of cinema, motion capture is certainly one of the most important. However, the use of this new technology has brought with it new challenges for the actors, who found themselves having to act in a suit, without any scenario, interpreting characters of appearance and movements almost always different from their own, and to having to imagine the whole scene in their mind as they perform it. With this thesis and therefore with the use of AR for motion capture shooting the aim is to join this scenario of change in cinematographic production given by technology, offering a way to help the actors in shooting this type of scenes, decreasing their workload, especially in terms of imagination, and helping them with the other difficulties they face when they have to perform scenes with motion capture. In the three sub-chapters that follow, the production pipeline of a film that includes special effects such as motion capture and how this pipeline has changed with the advent of new technologies will be first analyzed; then, the focus will be shifted on how a motion capture shoot is made, and finally on the difficulties of the actors in having to deal with scenes that require motion capture.

1.3.2 The VFX pipeline nowadays

Making a film is a difficult thing to do. Making a film which includes complex VFX can be even harder; a lot of different departments and people are involved: production department, virtual production department, art department, actors, director, assistants to the director, animators, grip personnel, editors, VFX supervisor, and so on. Let's now try to understand how a film production who involves VFX, and then more specifically motion capture, works. There are a lot of articles with examples of a VFX pipeline. Andrew Whitehurst, a famous Award-winning VFX Supervisor, proposes a pretty accurate typical VFX pipeline structure a web article [38] (Fig. 1.24).

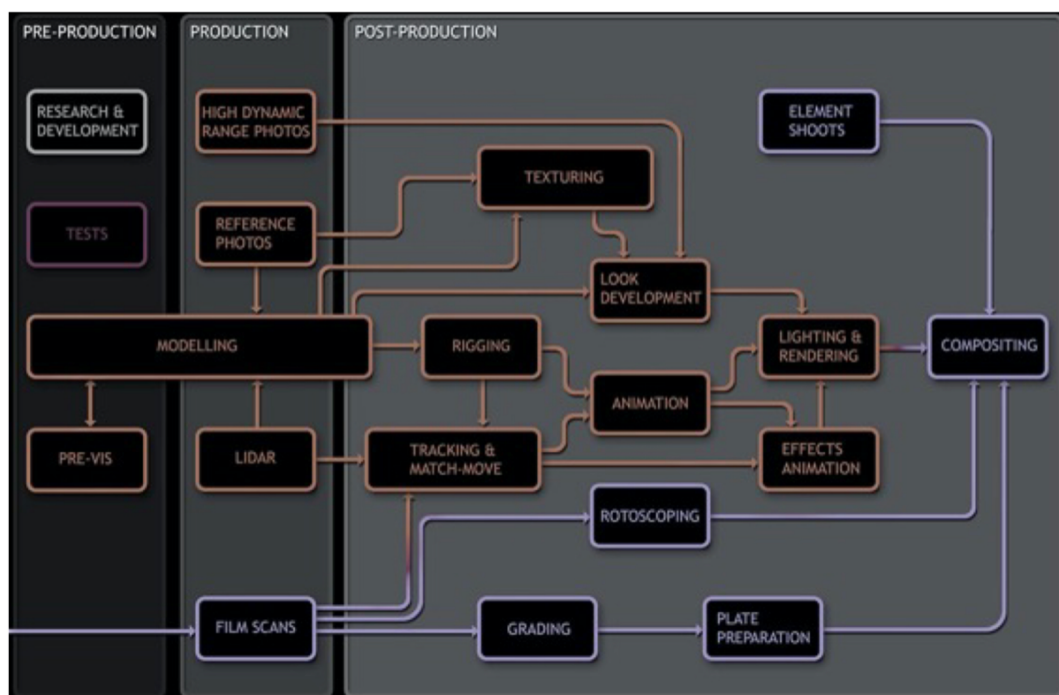


Figure 1.24: VFX films pipeline presented by Andrew Whitehurst [38]

Although the article is from 2008 this pipeline it's still a pretty good and accurate reference to start understanding the complex world of film production when virtual effects are involved. So let's start from the basis. The pipeline is divided into 3 phases pre-production, production and post-production. The pre-production phase include research and development, tests, pre-vis, and an early production of assets. In this phase preparation and planning are done. In particular, in the research and development period, technical approach to the film's effect is decided focusing on software preferences and techniques. Then tests are performed to demonstrate either a potential look, style or piece of technology in order to convince clients that

what shown can be used for the film. In the pre-production phase, as said, assets like background or characters start being created for the pre-vis. Pre-vis stands for pre visualization, and it is essentially the process of converting a storyboard and script into a 3D animated, low quality rough draft for each VFX shot. At this stage the director will get an idea of how the scenes will be shot, and can experiment on camera moves and set-ups before committing to an expensive on-set shoot. Later it will be shown how pre-vis came to be more and more important in the production of a film. The production phase is the stage where actual shooting of the film is done. On set, certain members of the VFX team are present to take as many photographs as possible to use them later on for modeling, texture, lightning references, and as possibly elements for matte paintings. LIDAR and other devices, instead, are used for 3D digital scans of environments, buildings but also for props and actors to create high detailed models that can be use later as references. High dynamic range photos are also shoot to use image-based lighting techniques when lighting the shots. If motion capture is involved also a specialized team is present on the set to arrange all the equipment needed (mocap suits, tracking cameras, head mounted devices for facial capture etc.). In the post production VFX and animation teams works to integrate the virtual elements in the film shoots. This is done in many phases. Rigging is what enables 3D models to move and allows to animate characters. LIDAR scans and camera tracking done in the production phase are used to add environments and elements and putting the CGI elements in the right orientation. Of course textures and effects like physics simulation are added but the most important factor to make the film more realistic as possible is the fidelity on lighting. At the end the final image is composed and the live footage, matte paintings, and various VFX renders are blent together. The VFX department at this point have finished his job and all the scenes goes to the director for a final color grading. These are the basis for understand in principle how a VFX film pipeline works. However, it shall be recalled that when used in actual production, a pipeline is rarely truly serial as this ideal pipeline. In particular in recent years the production of films with CGI elements is changing due to the various improvements of technology. This change is increasingly leading to a so called: “Virtual production”. Virtual production, as stated in [39], is a broad term referring to a spectrum of computer- aided production and visualization film-making methods that, combining virtual and AR with CGI and game-engine technologies, enables production crews to see their scenes unfold as they are composed and captured on set. Epic Games, in [39], made a guide to better understand virtual production full of interviews with directors, virtual production supervisor and other important figures in the production of a film. In [39] Noah Kadner shows the difference between a normal film production, like the one described at the beginning of this section, and a virtual production. The main difference between the two pipelines, showed in Fig. 1.25, is that, while the traditional one is more

linear and tends to keep the different tasks of the various departments separated, the virtual production involves various departments from the beginning increasing the collaboration between them and creating a more iterative process.

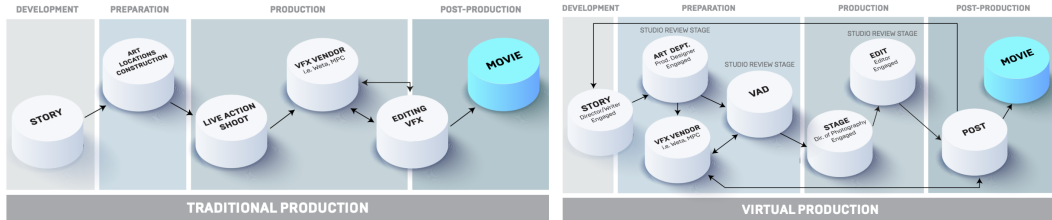


Figure 1.25: Differences between traditional and virtual production pipelines [39].

An important figure in this new virtual pipeline is the VAD. The Virtual art department (VAD) is in charge of developing real-time assets from initial look development to production-ready models and environments. The work of this department is generally focused on delivering complete camera-ready assets for production. Their primary role is to develop real-time assets from initial look development to production ready models and environments. Unlike the traditional pipeline in which art department had to create background, character, assets etc. without knowing how the final result on the scene would be, and VFX department had to add VFX just after the shooting, VAD is more involved and acts like a liaison between art department, VFX department and the shooting itself, handling a larger share of the finale imagery. With this new type of pipeline, more importance is given to the preparation (the pre-production) rather than production itself. With new technologies the pre-visualization has improved a lot. Ryan Stafford, an executive and VFX producer, in [39], claimed that when filming the film “Call of the Wild” they were able to pre-visualize the entire film, and could even show the previous version to an audience. This not only let the director and other important figures to understand how the film was going to be, and by doing so, sharing the same vision and avoiding all the possible misunderstandings of thinking of a scene from different perspective, but it was also a way to see the audience reaction, taking notes of what worked and what does not in terms of story but also shots or lightening etc. Wes Ball, a director, in [39], said that previs helps the crew to see the basic vibe that a director want to achieve, and so be on the same page about the project. Moreover, Felix Jorge, who worked in a lot of previs, in [40] said that now a director can even come, put an headset on and look at an environment in 3D made for previs, that is representative of what might actually will be in the final product. All of this because now they are able to create environment with a much higher fidelity than it used to be in previs in a fraction of time. This is just an example of how different figures (directors and previs department) can collaborate since the beginning in this new type of film production thanks to new technologies. Another advantage in

preproduction is that assets can be created with more visual fidelity and real-time efficiency from the beginning, and so they can be used from previs to final pixel. Level of detail control and decimation of assets is build directly into a virtual engine to enable using the same assets decimated enough to work in real time at the desired frame rate but still preserving visual fidelity. By decimation one means that the 3D object is decreased in complexity and polygonal resolution in order to meet the real time restrictions. It is worth to notice that nowadays losing polygonal vertices do not necessarily means losing too much of the final quality, in fact textures can be used to guarantee an overall good looking by means of bump map. This is the reason why more and more often the assets created from the previs team can be used also later on in the production without the necessity of recreating them (as was done before). For this thesis, knowing that the assets made in the early stage of a production (previs) nowadays are already done in high quality is important because it certifies that the AR-mocap system could be used already with more realistic 3D rendering making it more helpful for actors who could see the virtual character more in details. Other examples of technologies that are making the migration to virtual production possible are simulcams and led walls. Simulcams is a system that compose in real time virtual elements creating a live action frame. The simultaneous visualization of live action an CG characters in camera enhances filmmakers ability to optimize shot framing and overall composition. This system has been already used in a lot of film productions, like “Avatar” (2009) or “Rise of the planet of apes” (2011) for example, also when motion capture was involved; and gives the camera operator and director the ability to get an understanding of what the shot’s going to be like. Fig. 1.26 shows an example of simultaneous visualization.



Figure 1.26: Example of simultaneous visualization on the set of “Welcome to Marwen” (2018) [41].

Led walls, instead are giant screen comprised of interlocking LED panels used in films production instead of green screens to see in real time the virtual scenario and effects. Directors can also choose to change light setting or even the composition of the scenario itself in real time while shooting. But is not only useful for directors, everyone in the crew is able to see exactly what is in the shot. It is no mystery that for actors working with green screen has never been easy, and with led walls actors can now react not to marker representing an imaginary image but the actual final imagery live. LED wall gives everyone a clear visual context. Kenneth Branagh, a famous director, in [39] said that when he was shooting “Murder on the Orient Express” (2017), the projection screen, used to create the imagery outside the train windows, were critical for them. All the actors were galvanized and the image on the screens were so convincing that entirely transported the imagination of all the cast and crew, and even him, when walking into the fake train looking at led screens had to remind himself that he was looking to projected images. The use of this technology actually helped actors and crew to feel like they were on a real train and ultimately perform better. Two example of the use of ledwalls are showed in Fig. 1.27.



Figure 1.27: Two example of use of ledwalls: on the left a picture from the backstage of “The Mandalorian” (2019) [42], on the right a Sony Crystal-Led used on a photographic set [43].

So all of this technologies are used to let different departments and people involved more aware of what the final product would look like so that everyone can share the same vision of the final imagery. [39] state that the closer one can get to a shared vision that everyone can see and refer to, the more likely the final project will reflect that vision. In a recent interview, producer Ryan Stafford said that one of the things he always struggled with is getting the rest of the crew on board with what they are aiming to achieve, and with virtual production now he can. So virtual production is more and more used nowadays in production, by means of a lot of different technologies, it changes the traditional pipeline making all different departments more involved, aware of the final result, and connected to each other

from the beginning and increasing the pre-production work while decreasing the production and post-production work, making all the process more iterative and reducing overall time and costs.

1.3.3 Making a motion capture shoot

Now let's see more specifically what happens when a motion capture scene is involved in a film production. In [44] it is proposed a pipeline for VFX film production when motion capture is involved (Fig. 1.28).

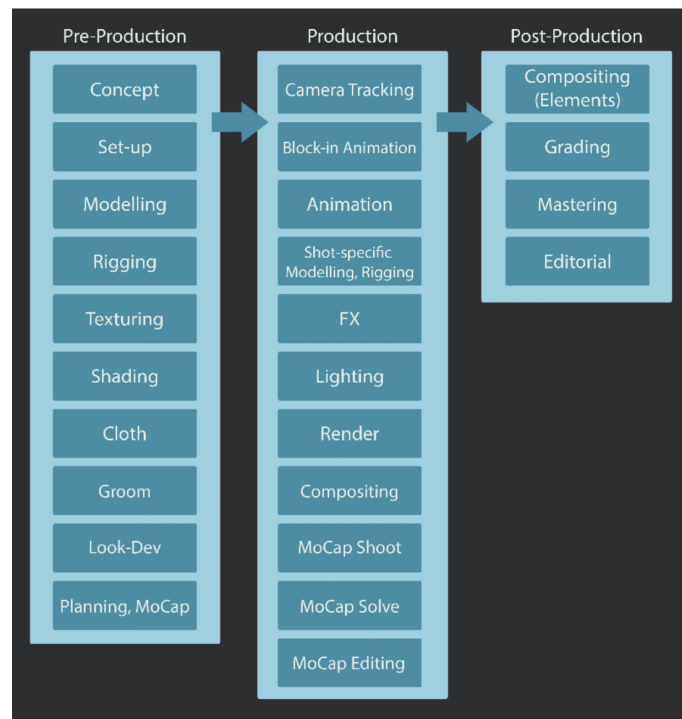


Figure 1.28: Example of pipeline for a film that uses motion capture [44].

As said in a previous section the assets are all made in pre-production to be used when shooting, and, because of the technologies improvements mentioned before, some of the post-production phases in this pipeline can be done on set. Speaking about the assets, for mocap session, also rigging is done in the pre-production to allow the real time animation of virtual character while on set in order to see it through the simulcam. As said in [45], often characters are based on the look, movement, and physicality of the actors playing them and, in this cases, usually, in pre-production, scanning, facial survey, measurements and photography of the actors are used to create the virtual character asset. From here, characters are conceptualized and motion capture puppets created. Of course also props, set decor,

lighting set-ups and other assets are created to be used in the shoot. Because often actors play virtual characters which are larger or smaller than them, in this cases the physical world has to be scaled appropriately and tracked against the stage take that contains the performance. This is useful when physical props that has to be tracked are involved in order to create them the right size. In pre-production is the production team who break down the previs, storyboard and editorial references into sequences, scenes. Preparation of a shoot also includes assign names to actors, stunt actors, and anyone/ anything present on set, even witness cameras or face mounted cameras and head gear used if facial capture. This is done for keeping track of all the elements on set and this is very important in productions. It's worth to notice that shooting with motion capture is a quite expensive and so everything must be planned in the better way to avoid useless wasting of time when actually on set. So, after the virtual department has created all the assets needed (even involving the actors as references if needed), and the production has planned all for the shooting, providing all the technology needed for the motion capture and organizing each element on set in the best way, finally all the crew, including actors and directors obviously, are ready to come on set for the shooting day. On set the keyword is flexibility, the virtual production team should be able to react quickly to the request of director, DP, editor, VFX supervisor: technology must be subject to the creative will of the artists allowing them to do quick changes if needed. There are a few technical things that has to be done in preparation of the motion capture shoot. One is the camera tracking. The camera tracking is needed to connect the real camera to the virtual one in order to visualize live the virtual elements of the scene, in our case the virtual character. This is useful, as mentioned before to have an overall look to the scene with real imagery combined with virtual elements through a simulcam. Another important thing is obviously setting all the motion capture system, and this includes mocap suits on the actors. The motion capture team help the actor get dressed with the mocap suits, and place optical markers (if optical system is used) or take care of the inertial sensor (if this type of mocap system is used). After this, actors must perform some standard poses and movements in order to calibrate the motion capture system (generally T-pose or A-pose). Fig. 1.29 shows an example of T-pose calibration. After doing this the movements of actor/actors are captured and transmitted on computers in forms of 3D points (if is an optical system) or motion data (if is an inertial one) and the solved onto virtual characters who now moves just like the actor.

Now, usually some rehearsal with mocap suits on are performed, and by means of all the technologies we've mentioned in a previous section the director is able to see in real time what the scene would be like, with CGI elements, lighting settings ecc and make his thoughts and adjustments if needed. Wes Ball in [39] said that now having all the assets ready, on set, just bringing a camera and an actor with the mocap suit on to do some rehearsal, he could "see the film" with a sense of how



Figure 1.29: Actors standing in T-pose for the calibration of the mocap suit in the making of “The Witcher 3: Wild Hunt” (2015) [46].

he wanted to block and shoot scenes. Anyway, how to do rehearsal and the time used on shooting a single scenes depends on the director. There’s not a precise method for preparing a single shot, for example Kubrick was known to spend even a whole day on just a single scene. The director decides how is better to prepare a single shot for actors, deciding on the basis of how he or she sees the scene, his performance considerations, the interpretative ability of the actors, the technology available to help them understand the scene, his will to try different possibility of the scene ecc. So, while shooting, real-time 3D visualization is rendering out and, as said before, with new technologies like simulcam, a preview of the finale scene including virtual and real elements is available in real time. Now, as explained in [45], favored performances are noted and made ready for the next phase. At this stage little changing on assets can be made, and, if needed new props are created on the fly, taking care if naming conversion, rigging etc.

Editorial has the task to take all the media from production like witness camera footage, real time render from virtual production team, sound, and the playback rushes from video assist and cut them together in a performance assembly. In Fig. 1.30 it is possible to see a frame of a performance assembly provided by the editorial. This performance is the chosen one from stage and informs other teams of what is needed along with confirmation that all assets are sync and correct. The performance assembly so becomes a task for virtual production, that has to produce a polished real time version of the scene. This tasks that were usually



Figure 1.30: Example of frame of a performance assembly provided by the editorial after a VFX shot production [45].

done in post production (as can be seen in Fig. 1.28) now are often done on set as the shoot commences and are also executed once the performance capture shoot is completed (as stated in [45]). In post production, finally, motion capture data is cleaned up and made final for the body animation of the character and props. This process can include sticking different performances from different actors or the same actor in different takes to make the final scene. Often animation adjustments and additions are required to improve the character's performance. Then the 3D animation of the motion capture character is completed and can be used by director and VFX for cut and VFX. This is essentially what happens on set when motion capture is involved.

1.3.4 Actors challenges with motion capture

Now let's spend some time analyzing the motion capture from the point of view of actors and directors trying to better understand their difficulties and their possible needs. Motion capture gives the possibility to actors to make an unlimited range of characters come to life, not necessarily with a human shape. For example Benedict Cumberbatch in "The Hobbit" (2012) starred has a dragon, so his movements were mapped on a very different shaped character. This lead to a complexity for the animators who have to bond correctly all the tracked points between the actor and the character, making the dragon (in this case) as similar as it can be to what

the actor wanted to be when he or she performed, but also for the actor himself who has to empathize with strange fictional monster, thinking of their real body as the character's body and give a "soul" to the virtual character only through his movements. So the motion capture performances are very different from normal performances: the actors have to think different, imaging of being not only someone else but also something else from what they are while performing in terms of body shape, emphasizing the movements and the gesture to produce better date, and most of the times imagine the scene in their head cause in the major of cases motion capture scene are filmed in big empty room. In a recent study [47], R. Ge and T. C. Hsiao explored various way for VR, AR and MR to help the film and television creative industries, and, talking about an actor performing a scene in which a giant snake (made in CGI) has an interaction with him, they stated that if the actor could really see the snake, the feelings of fear would certainly be more real. Richard Dorton, who works in the cinema and video games industry for film motion capture scenes (he worked in "Spiderman 2" (2002), "Star Wars: the Force Unleashed" (2010), "God of war" (2018) and hundreds of other films/video games productions), also known as "the mocap man", has even created a school to teach actors how to perform a scene that requires motion capture. In a recent video for Insider [48] he says that "motion capture is all about pure imagination" and that's the most difficult part for an actor. Moreover, in a recent speech at GDC talk (2017) [49], Andrew Ray, an actor, talked about techniques developed over time to improve motion capture performance, and most of them are different from normal acting techniques. For example he says that to play characters in mocap who haven't got the same body shape of the actor and make it "come to life" through the actor's move, it is very important for the performer to find a new natural state of balance of himself which matches the natural balance state of the character he's playing, physically. The actor does not know how the character he or she playing looks while playing it, (in most cases he or she is just told by the creators about how they have imagined the character ad maybe how he or she looks like by seeing some concept art) but he or she knows that his personality and characteristic will be shown only through his moves. And is not a very simple thing to do. So acting became more difficult when virtual elements are added in the scene, environment, characters played in motion capture or other elements and this is the reason why in the past years production have found ways to help actors and directors performing this kind of scenes. For example while performing for the video game "Hellsblade: Senua's sacrifice" (2017) actors could see in real time the rendering of the scene they were acting, better understanding the final result (Fig. 1.31). This is useful not only for actors but even for directors, who can see in real time the scene with all the visual elements in it. Also, in the film "Rise of the planet of apes" (2011), all the apes were made using motion capture suits and facial tracking, but while the other actors who acted in the scene could just see the performer with the suit

and the helmet with the camera for facial tracking acting like an ape, the director in camera could see in realtime the actual 3D virtual ape instead of the mocap performer (Fig. 1.31), and this has been very useful for the director Rupert Wyatt to better understand the scene and give important advices to the actors [50].



Figure 1.31: On the left two actress helped in acting in motion capture by a real time view of the virtual characters they were playing in the making of “Hellsblade: Senua’s sacrifice” (2017) [51], on the right an actor playing the role of a monkey in “Rise of the planet of apes” (2011) and the virtual result who could be seen live by the director [52].

Jerome Chen, a famous VFX supervisor, in [40], said that in a performance capture session he worked in, they let the actors see themselves as the virtual characters, in that case they were soldiers in an afghan tunnel, and state that that type of visualization enhance actors performances and give them ideas they might not have thought of otherwise. He said that it has a significant impact on the perform. Also, when filming Ready player one (2018), a film in which a virtual world is simulated, to Steven Spielberg, the director of that film, was given an head mounted display to enter in the virtual world that he was representing in his film [53]. Through this he could actually see what the spectators would see, but most important what the characters in the film would see to understand what they would feel in this virtual world. Moreover, it was given to him the opportunity to place a camera in the virtual world and actually film from there, understanding, in this way, better where to place the camera and other important aspects for a director’s work. Spielberg also wanted the actor to use the head mounted display and spend time in the virtual world their character in the film are, so they can better emphasize with their characters and feel in some way what they would feel in the film. It was very important to him that they understood what it was like for their characters to be in that virtual world so they could perform their scenes better. This to say that in film production it is usual to try to find ways (often using technology) to better deal with scenes that involves virtual elements and to overcome in someway the difficulties that actors and directors have to face. However making actor play in motion capture is not only a challenge for them but

also for other actors who are in the same scene with them. In a recent interview, Wes Ball, a film director known for the “Maze” trilogy, says that he talked with famous actors who were too intimidated by the concept of not seeing another actor in front of them with a costume but in a “motion capture pajama instead”. He also added that actors need help to visualize what the scene is, but it works when it’s about making a truthful emotional connection between characters. In fact for two actors who perform a scene together the most important thing is to empathize first with the other actor and then with the scene itself. They have to create connection with each other, and get along emotionally. This can be difficult when CGI is involved, and the other character is just a virtual one who’s added later on the scene in post production and so the actor has to play alone and only imagine the other character while performing the scene. Anyway in nowadays all productions try to avoid this type of situations, an example is the recent film “The Call of the Wild” (2020), a film about the adventure of a dog. Fig. 1.32 shows how the dog animation was done using motion capture.



Figure 1.32: Terry notary playing the part of the dog Buck in the film “The Call of the Wild” (2020) [54].

In this film a real dog was hard to use because of all the interactions that he has with humans (a real dog would have been difficult to manage in this situation), and so they used a human actor (Terry Notary) to play the dog Buck using motion capture. The producer said that this choice of using a human actor in a suit to play a dog could have been potentially silly, but it turns out that it was actually perfect for actors to play the scenes because they had someone to interact to emotionally, even tho it was a person and not a dog (“Terry’s performance has improved every actors performance in the course of the film because otherwise this actors would just be acting in front of an empty space” said the producer Erwin Stoff [55]). Harrison Ford (who also played in this film) says it was important to have an actual actor playing the dog to perform the scenes with him in order to understand where to look but most importantly to have someone to participate with, emotionally, even tho at first it was a bit challenging for him. Also in the film “The Call of Wild” (2020) Sam Rockwell played the role of Ivan, the gorilla, using motion capture and



Figure 1.33: Brian Cranston and Sam Rockwell on the set of the film “The One and Only Ivan” (2020) [34] while performing an emotional scene between a real character and a virtual character animated through motion capture.

his presence was crucial for other actors, like Brian Cranston Fig. 1.33 to perform some emotionally intense scene that otherwise they would have shot alone having to imagine the virtual character with whom they interact in the scene and which would have been added later in post production. This just to say than empathy and emotional connection between actors are important things when talking about actor performances and we must remember of them when we discuss about new systems to help actors in their performances. Another thing that productions does to help actors perform with virtual characters played using motion capture (and also to reduce the animators work and make adding CGI character more simple), is to use gimmicks to make an actor playing a character as physically similar as possible to the character himself. An example of this is the use of the face figure of the already mentioned character Thanos, in “Avengers: Endgame” (2019), or the costume of the beast in “The Beauty and the Beast” (2017).



Figure 1.34: Two examples of gimmicks used to help actors in the making of scenes in which motion capture is involved, trying to look alike the character they are playing in height/sized. On the left Josh Brolin playing Thanos [56], on the right Dan Stevens playing the beast [57]

Director Kenneth Branagh, in an interview for Epic Games [39], told that, when they were shooting “Artemis Fowl” (2020), they had some scenes where characters had to interact with a giant creature 4 meters tall and in that situations they’ve created a three-dimensional model of the creature and show them to the actors. This, as stated by K. Branagh, gave all the actors a sense of scale, a sense of bulk, and was inspirational, he said, for the actors playing the characters who were interacting with the creature and also for the actor who would be part of the motion capture element of giving the internal performance. He said that making sure that actors have the maximum of information as possible makes a significant difference to the way they perform. In this case AR could be very useful because it could let the actors see the giant creature by means of AR headset, also animating it in real time through motion capture. So for an actor perform with other mocap-suited actors can represent a real challenge, they have not only to imagine how’s their character feeling and imaging the scene in their mind but also to imagine that the actor with the mocap suit, in front of them is something else and interact with it in the best way possible, trying to emotionally connect with him, and this increase the amount of imagination work an actor has to deal with. It would be incredible if in this case Emma Watson, turning her head to the beast could really see the beast walking with her, or Dan Stevens watching himself could see a body of a beast Fig. 1.34, and, in general, if in some way actors could see in real-time the virtual characters in AR (either if they are playing it or other actors are) while they are rehearsing or performing the scene. So these are the main challenges that actors faces when performing scene in which motion capture is involved, and this is why AR could be really helpful for them to overcome this difficulties, reducing the amount of imagination work load, and finally perform a better scene.

Chapter 2

State of the Art

2.1 Introduction

Since the purpose of this thesis is to create a system that allows the actors to see in AR virtual characters animated through motion capture, the state of the art will cover three main aspects: motion capture, AR and the use of immersive technologies with the aim of helping film productions. So the related works that will be analyzed are those in which one or more of these aspects is present.

2.2 Related Works

Let's first analyze some works concerning motion capture. An interesting use of motion capture and VR is ImmerTai [58]. Chen et al. developed a system in which a user learn Tai Chi with a virtual instructor using HMD to live a virtual experience. The professor avatar performs the motion that has been previous captured, using a motion capture system, by a professional tai chi expert. Thus the user in the virtual world see his or her own avatar, the professor avatar, and the expert video, and try to replicate his or her tai chi moves (Fig. 2.1). The system then evaluates the user improvements comparing his or her moves to the ones done by the professional expert and gives him or her a learning evaluation. This allows the user to have fun and actually learn and continuously have feedback of what he or she is learning. The technology used in this work for tracking (either the expert and the user) is the kinect sensor, so the tracking is optical and this leads to a series of limitation such as low accuracy and poor stability, while also occlusion problems can occur. Due to this and other factors, another work has been done for learning Tai Chi in 2019 [59]. The idea is pretty much the same: the user with HMD Htc Vive "enters" in the virtual world where an avatar with pre-recorded moves (pre recorded with a Vicon mx optical motion capture equipment) done by

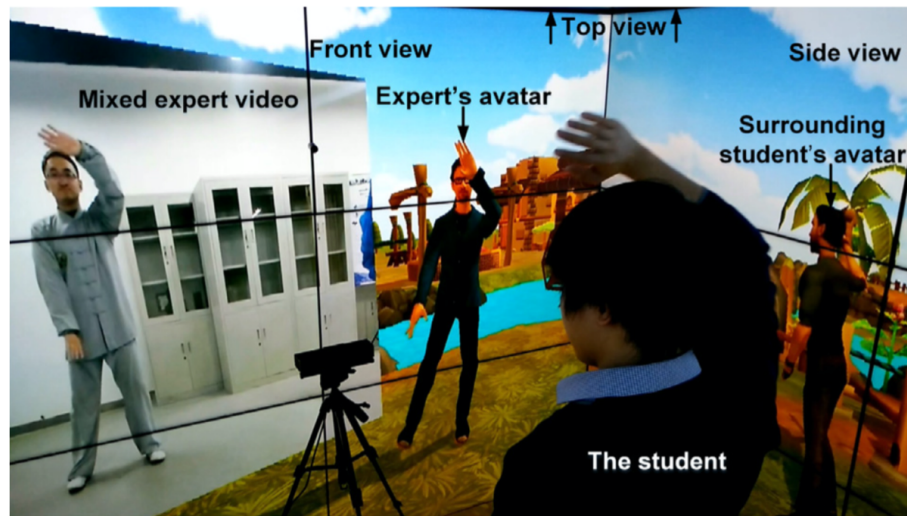


Figure 2.1: Student learning TaiChi using Immertai [58].

an expert shows him or her the moves, the user see also his or her avatar and has a direct feedback of the joints position in comparison with the avatar ones; the user can also move in the virtual scene to see the avatar from more perspectives, but in this case the motion capture system used implements inertial sensors (Noitom).

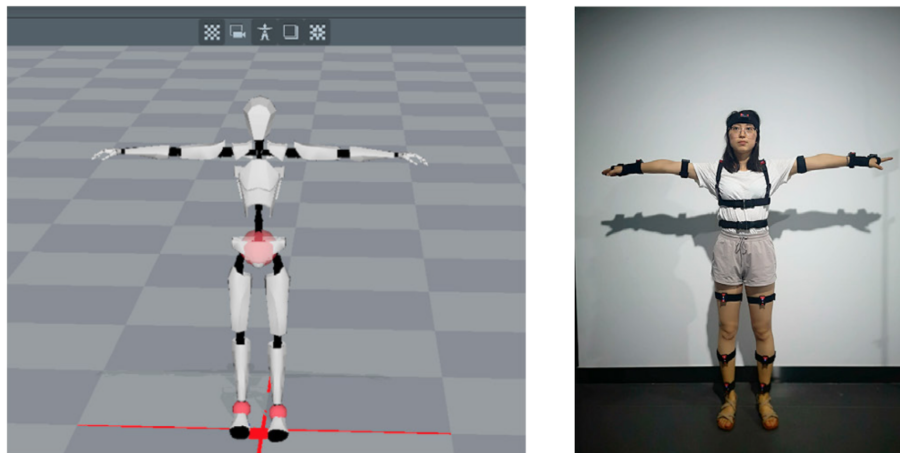


Figure 2.2: Woman wearing inertial motion capture sensors to use the virtual learning TaiChi system [59].

The user places on his or her body 17 inertial sensors, as it can be seen in Fig. 2.2, each one of them integrate an accelerometer a gyroscope and a magnetometer and, by this, the system acquires the position and orientation of the learners bones. The use inertial sensors allows to avoid the problems mentioned before that

can occur with motion capture optical systems such as lights or body occlusions. These are two examples of the use of motion capture for VR purpose, however in this works the expert moves are pre recorded and so the expert cannot give real time feedback to the learner: all the feedbacks are demanded to the system. In [60] instead it is proposed a real-time system for remote posture guidance for sports and physical activity training. In this scenario either the teacher and the student wears an HMD (Oculus headset), can see each other moves from the other one's perspective and can interacts in real-time. The movements are captured by a Kinect sensor like it was done in [58], and a current feedback is given to the learner. It is important to notice that in real time scenarios (like the one that concerns to this thesis) there are other challenges and limitation to deal with, most of them involving the transfer of the data and the rendering process. In fact, in the considered work the authors minimized the computation for rendering using stickman as avatars.

Anyhow these works only included motion capture and virtual reality; let's now see examples of the use of AR and motion capture together. Ikeda et al. in [61] proposes a self sport learning system to learn golf in which the learner with an HMD (HoloLens AR HMD) and a inertial mocap suit (Xsens Mvn) sees in AR the teacher's avatar (with pre recorder motions) and his or her avatar and receives feedback about his or her movements, as showed in Fig. 2.3. It is similar to what [58] and [59] have done with Tai Chi, but in this case the user sees avatars come alive in his or her own physical space by the use of augmented reality. This leads to all the troubles that AR can have, like occlusions, as said before, or the correct positioning and stabilization of the avatars in the physical space. These problems are related to the environment perception that HMD has of the physics world; it is worth to say that in the past few years a lot of improvements in this field has been done and now, the most recent HMDs have spatial awareness implemented in them and so they can understand better the real world and create a 3D representation of the surfaces in the environment.

Another paper in which motion capture and AR are both used is [62]. The setup is similar to [61]: an AR OST-HMD (Vuzix HMD) and an inertial mocap suit (Xsens MVN) are worn by the user. The goal of this project is to create a 3D avatar of the user that is perceived by the system and can interact with other virtual object. The position of the avatar match in every frame the real position of the user, and the avatar is rendered in real time on AR HMD considering the HMD position and orientation. The system takes the information about the position of the bones by the mocap sensors and the position and orientation of the head, which are needed for the correct rendering of the image, by the HMD. The authors use Unity 3D to process all of this information and create the virtual scene. In the virtual world in Unity, two cameras (for a stereoscopic view) are placed on the position of the eyes of the user so the system knows what is the user seeing



Figure 2.3: Using the AR HMD the user can see himself or herself playing with the golf-club and also the teacher avatar and his or her own avatar [61].

and know how to render the scene. In this way without using any markers or anchors they can place any object in the 3D virtual environment and their position and orientation will continuously update to match the user perspective. Moreover the absence of markers or cameras give the user more mobility through the space and this lead to a better sense of immersion. The authors affirm that the main limitation to this system is the limited FoV gave by the HMD, which however can be partially overcome using HMD with larger displays. This work in particular is very related to this thesis, because it involves a motion capture AR system and a 3D avatar whose position is connected to the real position of the person wearing the mocap system, and constantly updated in real time as the user moves, updating what the user sees through the AR HMD.

Now let's talk about some papers that presents works that involves augmented or VR and that are related to the cinema industry. A mentionable work is [63] in which Stamm et al. presented an Android app for low cost film production that let see, live, a virtual background (like instead of a green screen used for these type of scenes) while keep seeing the real elements over it (Fig. 2.4). They used Unity game engine to connect the virtual environment and what the user actually sees through his or her mobile device's camera. They were able to connect the position of the real camera and the virtual one by using Project Tango providing a live preview of how a final rendering would look like. This could be also useful for the actor playing in front of a green screen, because, connecting the mobile

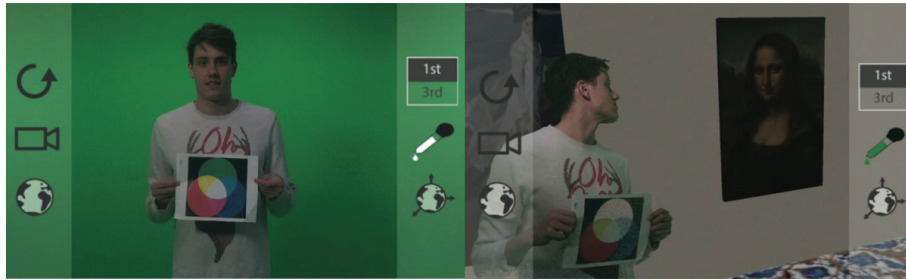


Figure 2.4: Through the app the user can see both virtual and real element in the scene [63].

device screen to a monitor would let him or her see live the overall scene with virtual elements already in it. This, as said in [63], “would give more believability to the actor’s performance, increasing his or her contextual awareness within the scene”, and that is exactly the purpose of this thesis too. Other works related to cinema industry that uses AR are [64] and [65]. These works proposes a real-time pre-visualization methods for visualizing virtual characters instead of actors in a real environment. Virtual characters are superimposed over the video in real time in live according to a real camera motion and an illumination condition. Both of the systems made this possible by using a pose camera estimation algorithm based on a reconstructed point based 3D model and finding 2D feature points frame-by-frame. Through this system, knowing where the real camera was, they could place the virtual camera in a correct position and so visualize the virtual elements over the real environment correctly. Moreover in [65], to improve realism, Tamura, T. et al. used photometric registrations to add shadows over the live video. In [66] Ichikari et al. went even further. Using the MR-PreViz system, already mentioned in [65], they create an MR action rehearsal system in which the virtual characters could be visualized through an AR headset in first person, and the person wearing it could even interact with them. In particular the scene they presented is a fight between two samurais, one real (the actor wearing the HMD) and one virtual as it can be seen in Fig. 2.5. They let the user interact with the virtual fighter giving him or her a motion tracked sword, so that, knowing the sword position they could understand when this got in contact the virtual one owned by the virtual samurai; a feedback is also given to the user through a vibration of his or her own sword.

This system was meant to be used by actors with inexperience in fighting action. The user wearing the video see-through head mounted display could practice with real-size CG enemies using his or her sword device as an interactive device, all of this while seeing the real world in background thanks to the augmented reality. This work, as this thesis proposed system, it is meant to help actors and uses AR through an head mounted display. This type of work could be helpful in



Figure 2.5: Actor using AR to rehearse a scene of a fight with a sword [66].

many different moments of a production. First, in pre-production, as powerful assistants to effectively visualize scenes that are not easily expressed (as a 3D MR storyboard); secondly for doing camera rehearsal and set simulation, pre-visualizing the scenes from more camera angles; and third even while shooting the actual scene: visualizing the MR rendering could be helpful to actors and staff as a reference or to share ideas. The pipeline stated in [65] about the use of the MR-PreViz technology is shown in Fig. 2.6.

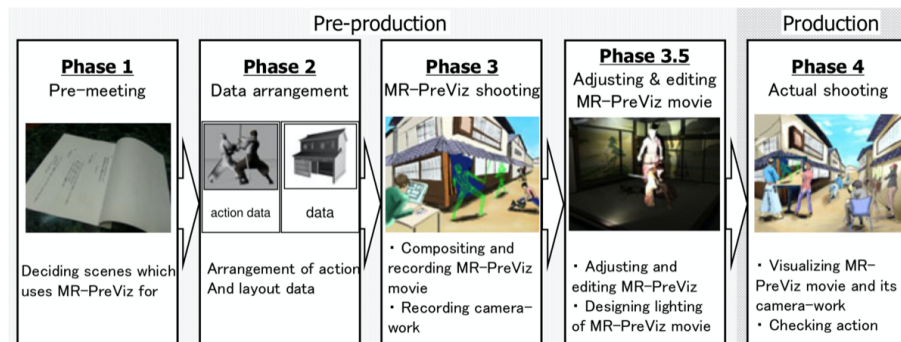


Figure 2.6: A pipeline that shows were the system presented by [65] can be useful in a film production.

It can be noticed that using this technology change a bit the traditional pipeline adding these new tasks (as choosing the scene in which use the technology, create action data, recording the environment, etc.) but in overall it could be very useful for the reasons mentioned before as for directors as for actors. It can also be useful in general to have more information possible about a scene before, on set production process, in order to minimize errors and reduce the need to re-shoot scenes later on in production. Moreover is good to have tools that gives to more people the ability to see a detailed preview of a scene decreasing the potential risk of imaging the same scene in a different way. Another mentionable work (even thought it

uses VR and not augmented reality) is [67], in which Berthelot et al. presented a system for help actor in their rehearsal, by the use of a VR HMD. Their aim is to replace classical actor's training for scene in which virtual elements are involved. In fact there are a lot situations in which actors have to perform in front of green screens or with partners that are only virtual, and so they have to imagine all of the virtual elements, which are added in post production, in their head. "Many actors complain about the frustration caused by the fact that they have to wait for the release of the film to see what they were really fighting against" says the paper. According to [67] the main problems when shooting these type of scenes are three. First of all positional problems: actors must be at an exact position or make a precise gesture while having no visual clue for reference (in general marks on the floor are used to help actors in this sense); secondly timing problems: VFX are usually designed during pre-production step so the actors have to adapt their performances by the timing of the VFX animation (for example when interacting with a virtual character); and third, gaze problems: when the actor has to follow a moving virtual objects, his or her eyes must point towards the object to follow (this is usually done by giving the actor cues for where to look). So in [67] the purpose is to help overcoming this problems using virtual reality. The system they presented not only let the actor see the virtual world and characters using VR headset, but also to interact with it. They support actors using two different systems. The first one is the "Interaction engine", through which the actor is able to interact with some of the objects that surround him or her; for example he or she can take a glass, move a chair, switch a light on etc just as he or she would have done in real life, making, as the paper says, the rehearsal more easy to remember for actors. The second system they have used is the "Scenario engine", which manage events triggered by the virtual environment or by the actor. This is very important for the actor to practice a scene more and more times, increasing the difficult of the scene: for example at first the user could let the events of the virtual environment depend on his or her play, and then try to perform the scene defining a timing that depends on the constraints given by virtual character's animations (which are done in pre-production): the virtual elements do not depend on the actor's play anymore, it is the actor who has to do a performance synchronized with the virtual elements animation in the scene. In the last part of [67], Berthelot et al. conducted a study involving 24 actors to compare classic rehearsal methods to their VR rehearsal for a green screen shooting. The actors had to perform three different actions interacting with a virtual tiger: 1) the tiger passes between the legs of the actor, 2) the tiger is sitting looking at the actor, and 3) the actor takes the orange back from the tiger. The actors had to train for these scene in the two methods (normal and using VR) and then fill out a questionnaire about their experience. The results showed the actors appreciated the VR method the most, demonstrating that VR rehearsal can actually replace classic rehearsal. The interesting thing is

that some of the actors involved in this study declared that experiencing the size of the tiger in VR significantly helped them to simulate the tiger passing through their legs, they found it easy reproduce same action during shooting and declared to feel more involved in the play. Moreover they commented that less concentration and mental engagement is needed when rehearsing using virtual reality, because they can feel more natural in their action rather than in classic rehearsal when they have to imagine everything. Fig. 2.7 shows the phases of the experiment done.



Figure 2.7: Three steps of the experiment in [67] to prove that VR could be useful for acting rehearsal.

All of this is particularly interesting for this thesis because it proves that for actors actually seeing virtual characters who they are interacting with in a scene, even if only while doing rehearsal, could be very helpful for them; citing [67]: “seeing and interacting with a virtual partner in VR and perceiving their size and stature makes it easier for actors to pretend they are having a conversation or that they are walking alongside”. Moreover, in the further works section, the authors also state that they can imagine in the coming years actors wearing AR glasses to rehearse, and this is exactly one of the use case of this thesis.



Figure 2.8: On the left two actors playing a motion capture scene, on the right the virtual scene in which they are immersed through their visors [68].

The last work that is worth mentioning is [68]. In this recent paper, Kammerlander et al. propose a system that uses VR to overcome the difficulty of the actors in shooting motion capture scenes in which they play two virtual characters who have a different size scales like the one showed in Fig.2.8. The idea is that by equipping both actors with a VR visor they are allowed to see themselves as a

virtual character and to see the other from the perspective with which their virtual character sees it. The tracked data is first captured by the optical motion capture system and the used by the application to animate the different scaled avatars (Fig. 2.9). The authors state that the use of VR would produce various advantages in shooting this type of scene. The first one is that actors acquire a greater sense of “body ownership”: they feel the body of the virtual character they interpret as their own, which is especially useful when this character is on a different scale from that of the actor and it allows the actors to empathize more with the character they play by improving the performance. The second advantage is that using HMD, the actors are immersed in the virtual scene and feel more mentally involved in the experience, and this decreases the imaginative work they are usually forced to do when shooting these scenes. And the final advantage is that the work done in post production by the animators decreases. This is because generally, by shooting these scenes without having an actual conception of the size of their virtual character, the actors do not position themselves correctly (for example with respect to the virtual objects that will be present in the scene but which they do not see in real life) or do not look in the right part (for example if they have to watch a virtual character who is smaller or bigger than them) and it is up to the animators in post production to manage these situations. On the other end, when shooting the scenes in VR, looking from the point of view of their virtual characters, the actors are able to position themselves and direct their gaze more correctly. However, using VR also has a disadvantage. Wearing the VR headsets, the actors cannot actually look into each other’s eyes, but only see virtual characters, which can make it more difficult to engage emotionally in the scene and empathy with the other actor, which can be fundamental especially in scenes more emotionally important and which is an aspect already mentioned in the previous chapter.

To evaluate this system, a study was conducted whose objective was to compare the results obtained by acting in a traditional way and those obtained using virtual reality. A script was created for a scene that the actors should have acted out. The idea was to write a scene that would bring out the advantages of using VR but that was not set too much exclusively on the aspects to be highlighted. The scene to shoot is that of a girl who finds herself in a room of her house in tiny dimensions and meets a huge monster, at first she gets scared then her interaction with the monster becomes more and more friendly and the two copy their movements, until she understands that is in a dream and asks the monster how to get out of it, and so the monster opens a door and she comes out of the dream. The scene leaves enough freedom for the actors, describing only the general context of the scene, however, it manages to show the advantages of using VR by 1) using two characters in totally different scales, 2) using a monster as a character, so it is important for the actor to identify with it and for the other to react as naturally as possible to its sight (which is aided by virtual reality), and 3) making the two characters copy

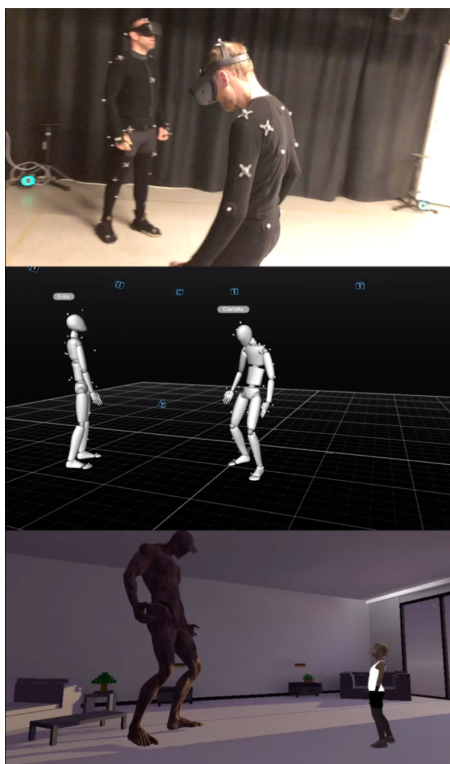


Figure 2.9: On top two actors playing a motion capture scene, in the middle the scene captured by the optical motion capture system, in the bottom the virtual 3D scene that they are seeing through their HMD [68].

their movements, which, if the scene were shot in a traditional way, could generate many positioning and animation problems. In this way the authors show how VR can therefore lighten the work of animators in post production. The study involved 22 people with different levels of acting experience and featured two execution modes: motion capture only or VR and motion capture. The participants tried both modalities, some starting from the first, others from the second, and then they had to fill in a questionnaire and then they were interviewed and asked for opinions and impressions. The questionnaire was based on three metrics: body ownership, social presence (how much the actors felt the presence of the other in the scene, which is important as it generates mutual understanding and greater coordination in the acting) and mental immersion (how much the actors felt mentally immersed and involved in their collaborative acting experience). The results of the experiment showed that while some actors preferred the traditional method of shooting the scene only using motion capture, mainly because it let them see each other in the eyes and make it easy to connect emotionally, most of them preferred the new proposed method that uses VR, and it also showed that the proposed VR setup

significantly improved the sense of embodiment and immersion over a standard mocap setting. These are the main papers that, for one reason or another, can be related to the subject of this thesis. Here they are inserted in Table 2.1 that shows for each paper what it has in common with this thesis work.

	Using augmented reality	Using motion capture	About film production
[58] [59] [60]	VR	✓	x
[61] [62]	✓	✓	x
[66] [65] [64]	✓	x	✓
[63]	✓no HMD	x	✓
[67]	VR	x	✓
[68]	VR	✓	✓

Table 2.1: Mentioned works characteristics.

So this thesis work is collocated among these works. In particular, it is related to [62] and [61] on a technological point of view: the use of motion capture in AR and, like in [62], the focus on the problem of collocating the virtual character in the virtual world as it can be seen through the OST-HMD in a specific real position. On the other side, speaking about an intentional point of view, it can be said that [66], [67] and [68] are very similar to this thesis because the aim is the same: to help actors to deal with scenes where they cannot see the virtual characters they are playing with in order to resolve some difficulties, e.g., concerning gaze direction or different scaled characters, and in the end acting the scene more consciously about the virtual elements present.

Chapter 3

System Architecture

3.1 Introduction

This chapter will outline the architecture of the devised system. The goal of the thesis work to allow a user with an OST-HMD to see in AR a virtual character whose movements are controlled by motion capture and is placed in the same real position as the person who pilots it via the motion system capture worn as is illustrated in Fig. 3.1. This type of work hides various difficulties. A first challenge is certainly to have the motion capture system communicate with the AR visor to transmit the tracking data, but the biggest challenge is to be able to make the best use of this data to position the virtual character in the correct position on the FoV of the user with the visor so that it is constantly in the correct position, superimposing itself on the person traced by the system that is piloting the animated body. All of this while trying to limit the latency as much as possible to have a pleasant result and not to create sickness problems to the user.

3.2 Basic Idea

The main problem, as mentioned in the introduction of this chapter, is to be able to show the virtual character in the correct position with respect to the position of the user using the viewer. As said, this can be defined as a reference system alignment problem, in that, the SRS of the motion capture system is different from the SRS of the HMD which is different from the real SRS. The idea to resolve this problem was to also track the HoloLens via Optitrack and use the information regarding its position/rotation in the Unity application in order to display the virtual character correctly. In this way it is in fact possible to represent both the position of the HoloLens and that of the avatar in the same reference system (that given by the Optitrack optical tracking system), and so the relative position between the viewer

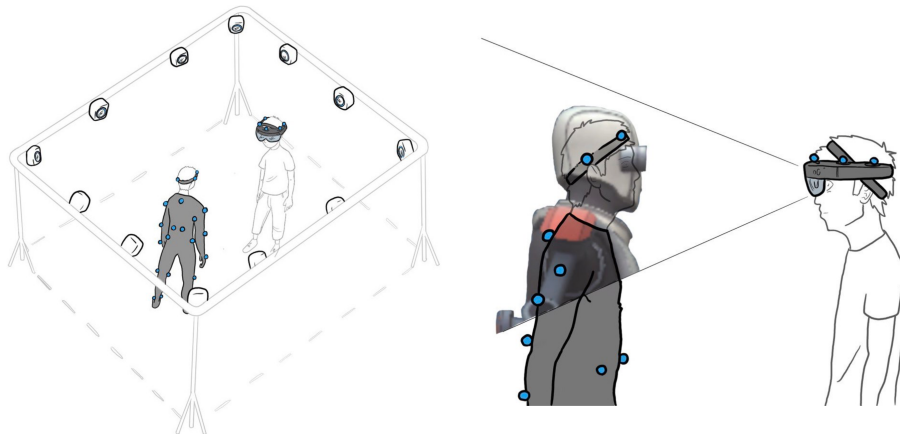


Figure 3.1: The concept of system functioning.

and the person tracked would be correct.

3.3 The MotionHub Choice

Initially, there were doubts about the motion capture system to use. The choice was between an optical tracking system, by Optitrack, and an inertial tracking system, by Xsens. The optical tracking system guaranteed less latency and guaranteed greater accuracy in tracking, however, it could only be used within the space seen by the cameras and can have occlusion problems while the inertial one, despite having higher latency and lower precision, could be used freely and without occlusion problems. In the end, the choice fell on the optical tracking system since, as mentioned, it was also necessary to track the viewer to allow the vision of the virtual character in the correct position and therefore it was easier to use the optical system as it allowed tracking the viewer simply by adding markers on it (Fig. 3.2).

However, despite having chosen Optitrack as a tracking system, in order to avoid the application, that would be placed on the viewer, to be strictly bounded to this motion capture system, an open source middleware called MotionHub [69] which is interposed between the tracking application (in this case the Optitrack software: Motive) and the application present on the viewer, was chosen to be used. MotionHub is a middleware that allows to receive data from a large pool of different body tracking systems (BTS), even of different types (optical, inertial, etc.) and even simultaneously, convert the data received into its own skeleton structure and in its SRS and then can send the data to a third-party application, in this case the one present on the HoloLens. MotionHub introduces a delay in data transmission to the application of only 13ms [69] therefore not influencing the



Figure 3.2: Optitrack tracker placed on HoloLens to be recognized from the optical system

correct functioning of the application. Using this middleware gives the advantage, as mentioned, of freeing the application to a single motion capture system and, because it is also an open source software, it could be modified, as needed, by adding new functions, e.g., a function to send animation signal to the application on HoloLens to start an animation of a virtual object in the scene at a specific moment.

3.4 Functioning

This section shows how the system works if only data coming from an optical system, i.e., the Optitrack, are used. Fig. 3.3 shows the main phases of the system.

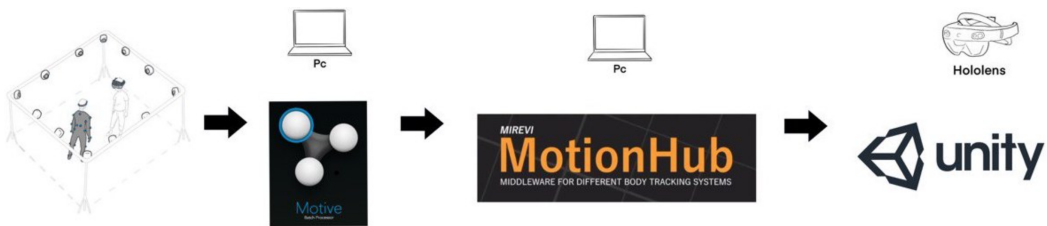


Figure 3.3: Pipeline of how the system works.

1. The data relating to the position of the suit and HoloLens are detected by the camera system and sent to Motive. To do this, as said, trackers have been placed on the HoloLens so that it too, in addition to the suit, is recognized by the Optitrack system which, in this way, can detect its 3D position in the

environment. The trackers placed on the HoloLens correspond on motive to a single RigidBody and therefore to a single position of the 3D space.

2. The data, via the NatNet SDK, are forwarded to MotionHub. Both the skeleton relating to the motion capture suit and an oriented point relating to the position of the HoloLens in 3D space as seen from the Optitrack system cameras are displayed on MotionHub as it can be seen in Fig. 3.4.

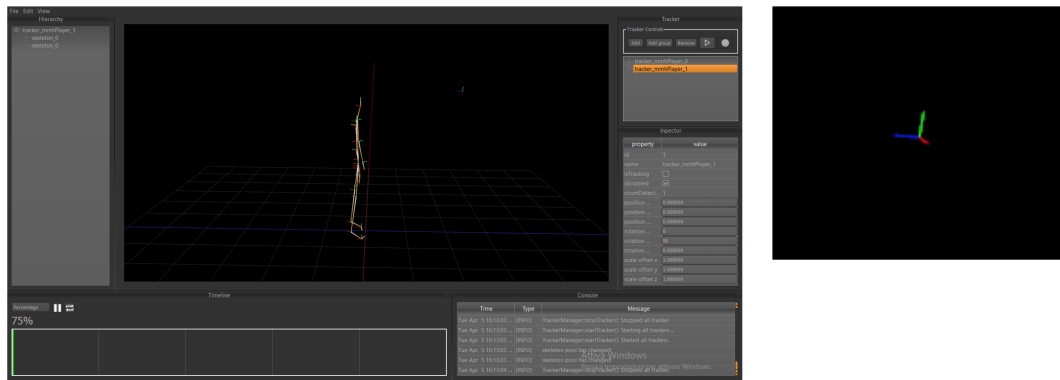


Figure 3.4: The MotionHub application receiving the skeleton and the visor data from Motive: on the right the visualization of the visor position and rotation data.

3. The data is sent via UDP connection, using the DatagramSocket class of the Windows.Networking.Sockets namespace, they are sent to the Unity application on the HoloLens. Then the Unity application receives two data packets: position and rotation of the HoloLens as received by the Optitrack system, as well as position and rotation of the skeleton joints that will be used to move the virtual character. The data is managed by the application.
4. While the avatar data are used directly, the data related to the position and rotation of the viewer as perceived by the optical system are used indirectly to drive the mainCamera in the unity application and therefore adjust the user's view and allow him to see the virtual avatar in the correct position as it is showed in 3.5. In Chapter 5 it will be explained more specifically how this data is used.

Therefore, summarizing: the data relating to the positions of the various joints of the suit and the HoloLens are taken from the Optitrack camera system, then sent to Motive (Optitrack proprietary software), subsequently forwarded to MoitonHub, which finally sends them to the Unity application running on the HoloLens that uses them to properly show the avatar in the correct position on the user's view.

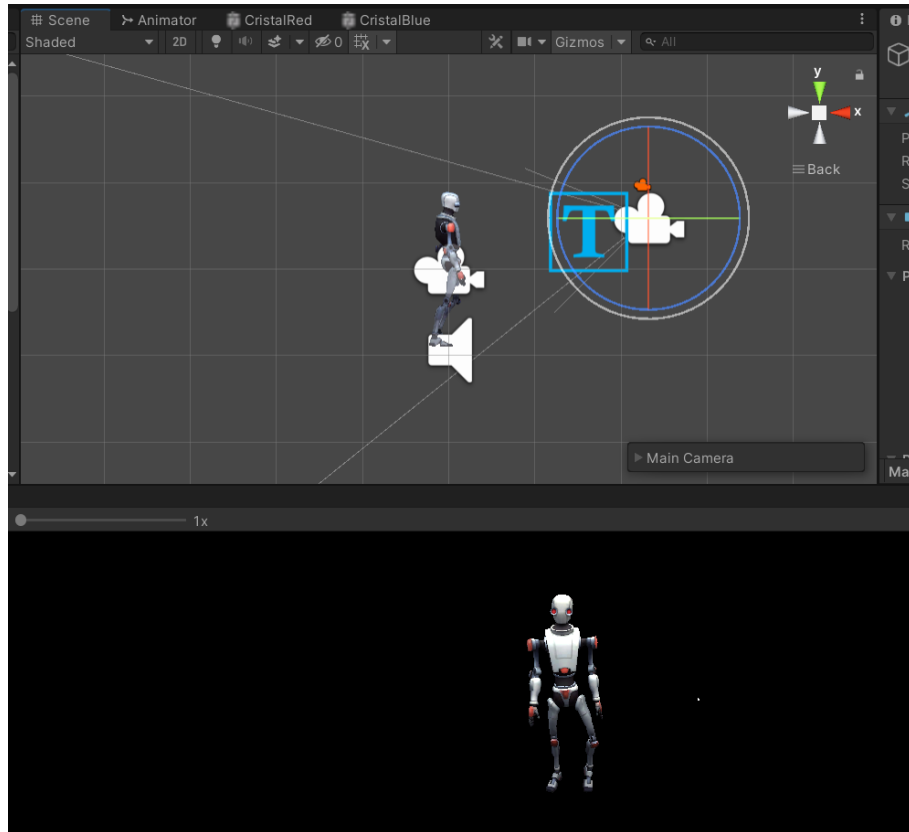


Figure 3.5: The Unity application receiving the skeleton and the visor data from motive and use it to move indirectly the main camera to change the view of the user.

Chapter 4

Technologies

4.1 Introduction

This chapter will show more specifically the technologies used for this thesis. Starting with the motion capture system used for tracking the data of the HMD and the person piloting the virtual character's movements (Optitrack), to the open source software used as a middleware to connect the motion capture software to the application (MotionHub), and finally the OST-HMD used to actually see the virtual elements in augmented reality (HoloLens 1st Gen.).

4.2 Optitrack

4.2.1 Introduction

OptiTrack system is a room scale infrared camera system that can be used to track motion data from body suits covered with markers as it can be seen in Fig. 4.1. Essentially, the hardware is composed by multiple synchronised cameras. These cameras are installed around the target capture volume so that markers reflect the infrared light emitted by cameras and 2D images are captured from each camera. Thus 2D positions are calculated, and the overlapping position data are compared to compute the 3D positions via triangulation. The data captured so is sent to Optitrack's proprietary software, Motive, that solves it, animate single rigidbodies linked to the markers captured or an entire skeleton if an entire body was captured and then can let you record it and/or stream it via UDP. In the next sections the entire system will be described more specifically.



Figure 4.1: On the left an example of Optitrack suit [70], on the right an example of an Optitrack motion capture setup with 16 cameras [71]

4.2.2 Hardware

The hardware used in this thesis is composed by eight special cameras positioned all over a $4\text{m} \times 5\text{m}$ spaces. Each camera is an Optitrack Prime x 13 camera (Fig. 4.2). This cameras are made specifically for an high speed precise tracking in a medium-size area [72]. Their size is $6.85\text{cm} \times 6.85\text{cm} \times 6.85\text{cm}$ they weigh 340g and have an invisible 850nm IR illumination. Because latency is something to deal with when talking about motion capture systems, the cameras used for tracking must have an high frame-rate. These cameras in particular have a 120 FPS high frame-rate and be used either with passive and/or active markers with positional errors less than $\pm 0.20\text{mm}$ and rotational errors less than 0.5 degrees. Passive markers, as said



Figure 4.2: Optitrack camera [72].

in the first chapter, are spheres covered with a retro-reflective material that can be tracked through infrared cameras, while active markers are LEDs that sends light signals to cameras by illuminating one LED at a time very quickly or multiple LEDs using a software to identify them by their relative positions. All the cameras are connected through a system networks that uses Ethernet cables. These type of

cables guarantee faster data transfer rates (1000Mb/second), provide power to all the cameras and they are also long (up to 100m) to allow the overall system to cover large spaces. A scheme of the cameras connection network is showed in Fig. 4.3. Before starting to capture all the cameras must be calibrated. To understand

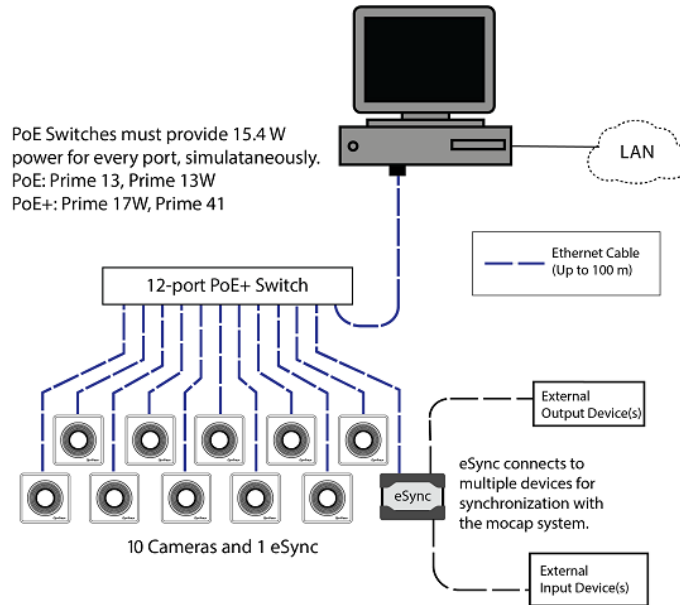


Figure 4.3: The Optitrack camera's connection via Ethernet [73].

why the cameras need to be calibrated one first have to understand basically how the cameras works. A single 3D point $P(x, y, z)$ can be represented on the camera using a projection matrix, so that this matrix determines how a real-world point gets projected onto the image plane of the camera:

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (4.1)$$

In this case f is the focal length of the camera. Anyway, the projection matrix for a camera is actually considerably more intricate. There are 11 intrinsic and extrinsic factors that characterize a real-world camera. The external parameters describe how the camera is oriented in relation to the outside world, while the intrinsic parameters describe how a world point is perceived on the picture plane. So, although P is still a 3×4 projection matrix, now there are 11 degrees of freedom, and not just the focal length as before. That is why a camera calibration is needed,

to determine these 11 degrees of freedom. In order to find the projection matrix of a camera, one must be aware of a point in space (3D) coordinates (X) and the matching picture point (x). So there are two equations to be solved with 12 unknowns. In order to find the projection matrix a “calibration object” like the one in Fig. 4.4 is needed. Essentially, the calibrator is a set of three trackers



Figure 4.4: An Optitrack calibrator [74].

whose actual mutual distance is known. Multiple synchronized cameras observe this calibration object (the three trackers) as it is moved in the space and use triangulation to associate the position in the 3D space to the ones in their projected image and find their projection matrix. The idea is that, if one knows the real relative position between three trackers and their position in the 2D projections of each camera, then there is enough information to calculate for each camera its projection matrix needed to capture trackers in every possible 3D position of the volume covered by the cameras. So in order to calibrate all the cameras, a person waves the calibrator (the three trackers) so that, after a number of frames, enough points to calculate the projection matrix are taken. So what happens after the camera system is calibrated to reconstruct a 3D point from 2D images? As it can be seen in Fig. 4.5, one must ascertain the coordinates of the X since each camera only knows the coordinates of the x . Projecting a ray through the center of the camera and through x , one may determine that X must be located along that ray. However one needs a second camera to determine the exact point along the ray. One can once more back-project a ray from the camera center through a point x' to X if a second camera observes the same location. Now there are two rays that pass through X , and their equations are known. One may get the three-dimensional coordinates of x and x' by locating the point X that fulfills the intersection requirement.

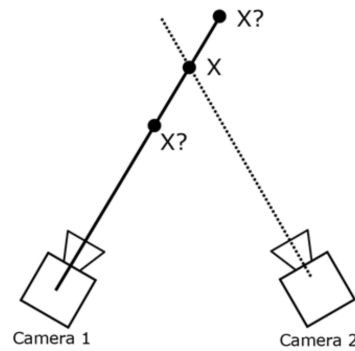


Figure 4.5: Two cameras watching the same 3D point [75].

4.2.3 Software

Once the data is captured by the cameras is sent using the ethernet cables to the computer connected and it's processed by the Optitrack software, Motive. This software basically processes all the camera data, find a 3D global position for each tracker, “solves” the skeletal data to give an hyper accurate tracking, and delivers all this data as global 3D positions, marker IDs and rotational data. Motive performs a continuous calibration so that, after calibrating all cameras once (as described in the previous section), it is no longer necessary to recalibrate them. Motive does this while data is collected during normal use of the system. The Motive interface is shown in Fig. 4.6.

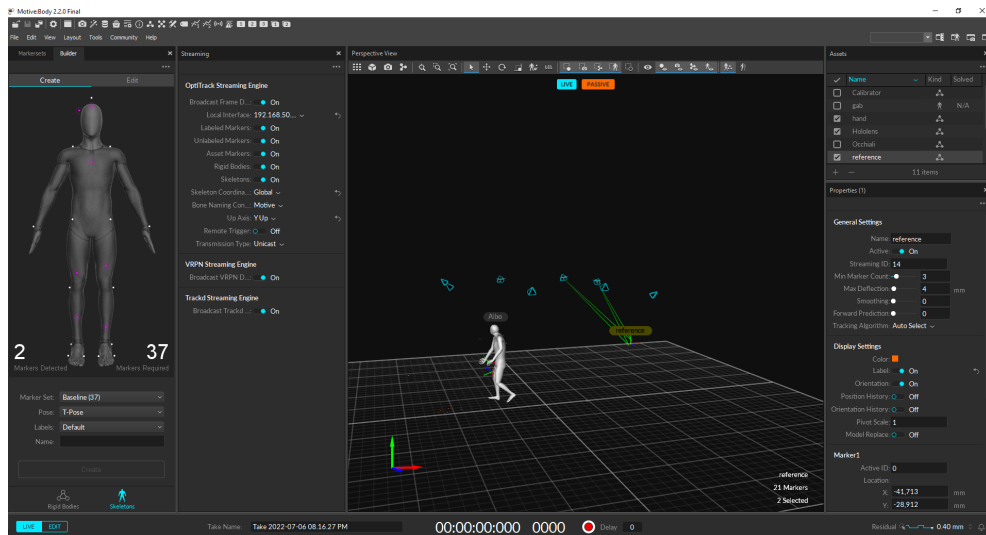


Figure 4.6: The Motive interface.

Basically in the 3D viewport the markers tracked by the cameras are visualized. The user can select more markers and create a rigidbody from them. A rigidbody is a unique oriented point in the 3D space defined by its markers. A skeleton avatar is composed of a series of rigidbodies, each defined by certain markers. Motive helps with placing correctly the markers on the person (who is wearing the suit) body showing an example of an avatar as it can be seen in on the left. To create a skeleton, when a person with a body tracking suit is visible by the cameras and so it can be seen in the viewport as a series of markers, the user have just to select all of the markers and then create a skeleton from them. However, as each person has a different body shape, the avatar created must be calibrated specifically for the person wearing the tracked suit. This can be made in various way but essentially the person must be in a specific position, like T-Pose or A-Pose and then the software, knowing that the person is in that position automatically calibrate the trackers to that specific person.

Then another manual calibration for each tracker can be made if needed. However after the markers skeleton raw data arrives to Motive is not applied directly on the avatar body but first it is cleaned and refined and then taken over by a solver. A classic solver is a process that calculate a pose with six degrees of freedom of each bone of the skeleton at each frame using the markers data. In Motive, the solver does more than that. The solver is precise and robust, and this means that it accurately defines the movement to give to the avatar piloted by the subject wearing the suits producing smoother animations, and it can also handle situations in which in one or more frames not all the markers that define the skelton are actually seen by the cameras, predicting their position and giving a correct character pose for those frames. Motive can also stream the data over a network. As it can be seen in on the left, there is a window made especially for this. The data is sent using the NatNet Software Development Kit (SDK). This SDK is integrated with standard APIs such as C/C++/.NET, and protocols like UDP, unicast or multicast, allows to integrate the Optitrack tracking data into third-part application for receiving real-time streaming. There are two formats of data streamed by Motive: DatasetDescriptions and FrameOfMocapData. The first one contains generic descriptions of all the data transmitted for the current frame. It contains all the skeleton description, the rigidbody description, the markerset descriptions but also server description, device description and so on. The FrameOfMocapData instead contains the data of a single frame. Each FrameOfMocapData referres to a particular frame and, in addition to information on markers, rigidbodies and skeletons in the frame, it also gives information on time and latency. Speaking about latency, the total latency of the optitrack system streaming data to a third part application depends on the latency of the cameras in capturing the 2D images, the software latency to elaborate the data and then the latency introduced by the streaming software NatNet SDK to stream the data over the net. Specifically, as it can be seen in

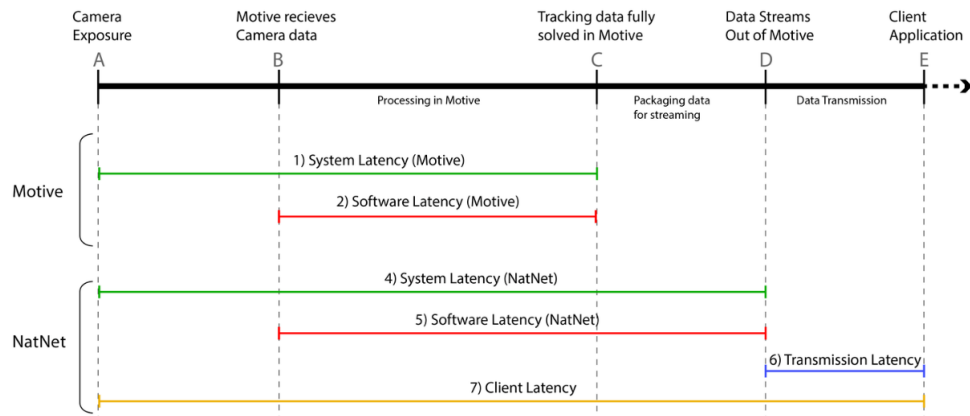


Figure 4.7: Optitrack stream latency [76].

Fig. 4.7, one can analyze latency from two main aspects: the Motive latency and the NatNet latency. The Motive latency can refer to system latency and software latency. The overall amount of time between when the cameras expose and when the data is fully solved is represented by the system latency of Motive, and its about 4.7ms. Instead, the software latency refers to the time it takes Motive to process each frame of captured data. This processing time includes the time needed to label and model trackable assets, convert the 2D data captured by the cameras into 3D data, display data in the viewport, and perform other tasks configured in Motive. Motive also needs time to convert solved data into the format required by the NatNet streaming protocol (which is about 0.2ms). Speaking about NatNet is latency can divided into System latency, software latency, transmission latency and client latency. The client latency represent the overall time since when the data is captured by the cameras to when is received by the client application. The time it takes Motive to process the captured data and make it completely ready to be broadcast out is represented by the NatNet software latency. The NatNet system latency instead refers to the time needed to the data to be ready for be streamed out since when is captured while the transmission latency is just the time need for the data to arrive to the client once is streamed by Motive. However the total latency of the Optitrack system from when data is captured by the cameras to when it reaches a client, as it is an optical system, is very low and is in the order of 50ms.

4.3 MotionHub

MotionHub is an open-source middleware that can receive simultaneously row data coming from different Body Tracking System(BTS) technologies, process it

to create skeletons with the same unified structure and then can transmit it to a third application client via Open Sound Control (OSC) [77] protocol. To receive data from a lot of different BTS, MotionHub used the respective BTS SDK, for example to receive skeleton data from Optitrack it used NatNet SDK that is the client/server architecture used by Motive (the Optitrack software) to send tracking data. But with different BTS comes different transmission methods and protocols, hierarchy structures, units, coordinate systems and different numbers of joints as well as rotation offsets between joints and so MotionHub is written to perform for each BTS a correct transformation of the raw data that produce for every BTS a skeleton with the same structure. This is the skeleton data structure that is then sent to the client application using the OSC protocol as said before.

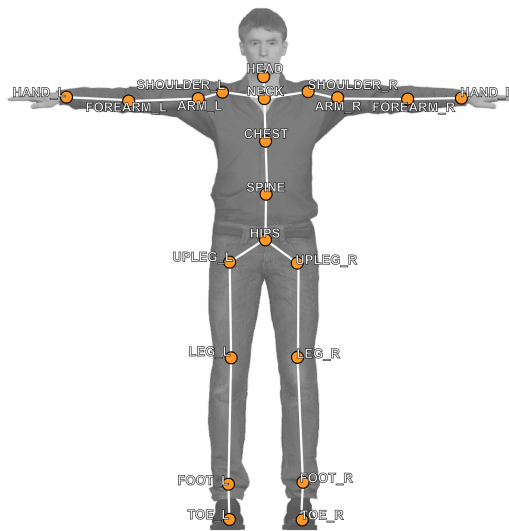


Figure 4.8: The unified skeleton structure streamed by MotionHub [69].

The data structure showed in Fig. 4.8 is based on the data structure used in Unity to animate humanoid avatars, so that is simple to use the skeleton data streamed by MotionHub in third part application created with the Unity game engine. Each of the 21 joints is represented by a global position (a `Vector3` in C#) and a global rotation (a `Quaternion` in C#). MotionHub uses multi-threading to process data from the various BTS system. So each thread receives the skeleton raw data coming from a particular BTS system, process it and then sends it to the client. Each BTS has its own capture frequency and update rate. MotionHub uses UDP because it has a faster connection and a lower latency compared to TCP. Each OSC packet sent from MotionHub consists in an address which is used on the receiver to understand how to use the rest of the message data, an ID of the skeleton which the data is referred to, and, for each joint, three values for position and four values for rotation. MotionHub's UI (Fig. 4.9) consists in a 3D scene to

visualize all the skeletons (converted into the MotionHub unified data structure) received from the various BTS, a Tracker Control Panel to manage the various BTS (add/remove/record etc) and a Tracker Property Inspector window to see the tracker's variables and set the offsets if needed. A Unity package as example of

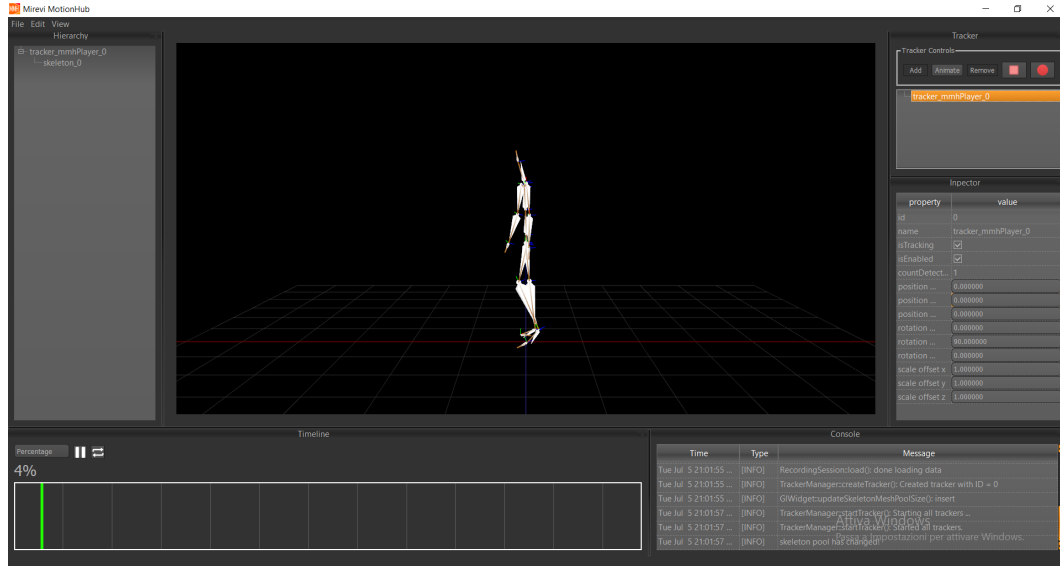


Figure 4.9: The MotionHub interface.

a client that receives data from MotionHub was also made. This package is able to receive all the skeleton data coming from motionHub and use it to animate a humanoid avatar. The first time a message with a skeleton with a new ID is received a new avatar is created. Then when a message with that skeleton ID is received the data is used to animate it. In particular the joint of the animator are positioned in the position received from MotionHub. To get the quaternion which represent the rotation to give to a particular joint, the avatar's joint rotation in T-pose and the product of all inverse joint rotations in the skeleton hierarchy above the present joint are multiplied by the transmitted rotation quaternion for all joints. The process iterates through the joint hierarchy upward, beginning with the parent of the joints and ending with the root joint. The character's local rotation is then adjusted using the product quaternion R. So, to summarise, motionHub uses specific SDKs, each in a different thread to communicate with different BTSs even simultaneously, then receives the avatar skeleton data and converts it in the appropriate way (different for each BTS) so that the received skeleton is represented in a standard way (with 21 joints like the humanoid avatars in Unity). It then sends this data using the OSC protocol that relies on UDP (preferred to decrease latency) and sends the data to a third-party application such as the one found in the Unity package they created, where position (`Vector3`) and rotation (`Quaternion`)

data is received for each joint of each skeleton and the avatars set in the application are animated. The data journey is showed in Fig. 4.10.

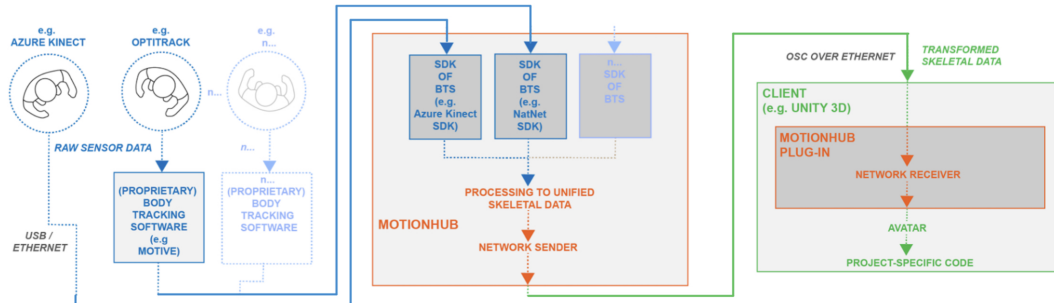


Figure 4.10: MotionHub workflow.

4.4 HoloLens 1st Gen.

The HoloLens is an OST-HMD released by Microsoft in 2016 (Fig. 4.11). Being an OST-HMD it is used, therefore, to project virtual images, called holograms in such a way that they are seen and perceived as being part of the real 3D environment around them. In this case, in fact, it is possible to speak of augmented reality or mixed reality, as the user, wearing the visor, can look at both real and virtual objects as if they were part of the same environment that surrounds him. For the user wearing the HoloLens, the holograms are perceived as actually occupying a specific position in real 3D space. In fact, as the user moves, he or she will still see the hologram occupying the same real position as if it were an actual part of reality.

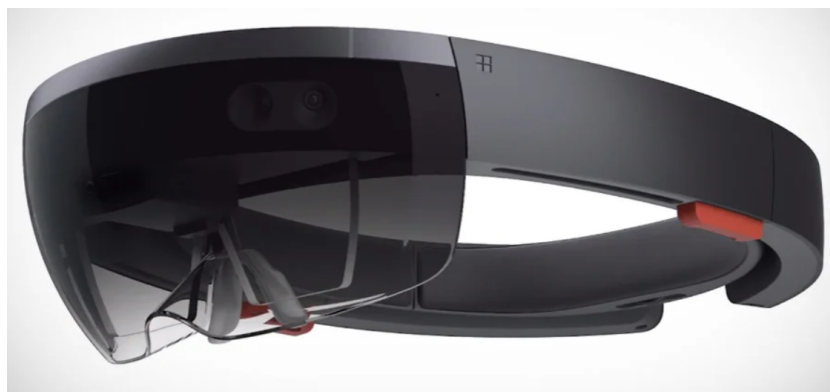


Figure 4.11: Microsoft HoloLens 1st Gen.

In order to wear the device, the user first secures the HoloLens on their head using an adjustment mechanism at the back of the headband that consists in a wheel that, if rotated, can widen or narrow the visor to the user's head supporting and evenly dispersing the weight of the device for comfort (Fig. 4.12).

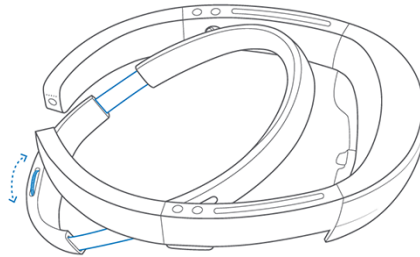


Figure 4.12: The headband with the wheel used to adjust the HoloLens to the user's head [78].

Speaking about the internal units, HoloLens is equipped with an Intel Cherry Trail SoC with a CPU and GPU, and a coprocessor built especially for the HoloLens by Microsoft called Holographic Processing Unit (HPU). Both the SoC and the HPU feature 1GB of LPDDR3 and 8MB of shared SRAM. As said, in order to function, an OST-HMD must have these two functions implemented: a conception of the surrounding space and the possibility of being able to display superimposed images on the lens through which the user views the real world. To be able to understand the world around it, the HoloLens comes equipped with an inertial measurement unit (IMU) which consists of an accelerometer, gyroscope, and magnetometer, and is situated directly above the bridge of user's nose on the holographic lenses, plus a depth camera, that is a particular type of camera that can perceive the distance to objects around it, with a 120° by 120° angle of view, a 2MP photo/HD video camera, an ambient light sensor and a four environment understanding cameras. The HoloLens sensor bar is showed in Fig. 4.13. This four cameras provide the basis for understanding the environment while the depth camera is able to perform spatial understanding and surface reconstruction which means that it understands the surfaces in the environment to let the holograms be able to interact with them. It is also important for hand tracking. In fact to interact with the HoloLens the user can use his own hands doing particular gestures, that can be seen in Fig. 4.14 that are recognized by this camera. In particular the bloom gesture by default is used to bring up the main menu every time is needed. The airtap is used to click, hold and drag an object. In order to visualize the hologram on the lenses the HoloLens uses basically two projectors, an optics, a waveguide, a combiner and some gratings to expand the image. Let's see them more in details. It all starts with the projectors that are microscopic liquid crystal on silicon (LQoD) displays positioned on on the lenses' bridge (behind the IMU) that shot out images



Figure 4.13: The HoloLens sensor bar [79].

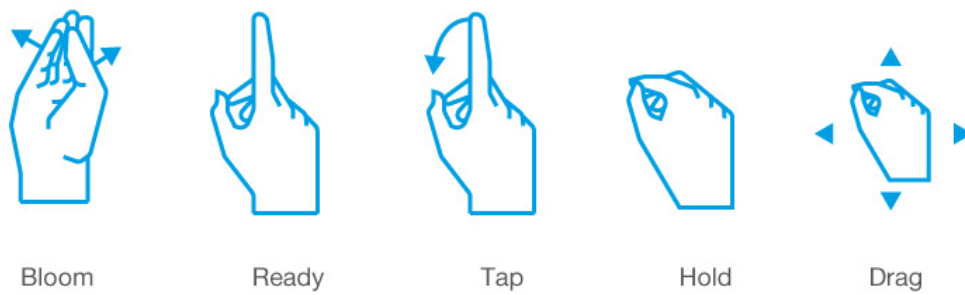


Figure 4.14: The hand gesture recognized by the HoloLens used by the user to interact with the device.

which then pass through a combiner. The combiner aim to blend the projected image with the actual environment. The HoloLens exploits an optical phenomenon called Total Internal Reflection (TIR) to project images onto the lens. This is an optical phenomena that happens when waves arriving at the interface of two media, are entirely reflected back into the first internal medium rather of being refracted into the second (the external), and that's what happened in the HoloLens where, because of the shape of the prism, the light coming from the projectors bounce internally on the lens and then comes out at the user's eyes. These lenses are showed in Fig. 4.15.

However there is a problem using this phenomena: the more the large you want the FoV the more large have to be the lenses in which the projector light bounce. That's why Microsoft uses wave-guides. Microsoft coated their surfaces with a material that enables the production of many de-fraction gratings, that essentially



Figure 4.15: The HoloLens lenses [80].

enlarge the image received by the user's eyes. These gratings are of different types to make RGB color holograms. Microsoft managed to equip hololens with a 30° FoV. So, to summarise, the HoloLens understands the world around it through a series of sensor consisting in cameras and light sensor, understand the users position and movement in the environment through an IMU, processes all this information using a CPU, a GPU and a dedicated chip called HPU, and thus understands where it must project the images onto the lens in order for the user to have a view of the hologram as if it were actually occupying a position in real 3D space. To project the image it uses two mini projectors that shot out the image to a prism called combiner that merge the projected image to the real world seen through it. Plus waveguides that cause defraction are used on the lens in order to obtain a larger FoV as possible.

Chapter 5

Implementation

5.1 Introduction

This chapter will talk more specifically about the implementation of the system. As previously said, the main difficulty of this thesis work was to let the user with the visor be able to see the virtual character in the correct position, superimposed on the person, tracked by the system, who is piloting the animated body. To do so, as said, the HoloLens position had to be tracked to get the relative position between the user and the motion captured body piloting the virtual character in order to place it in the right position on the HoloLens itself. So the HoloLens position, tracked by Optitrack, is sent to motionHub which has to interpret and then send it to the application that has to use it in the best way possible also in order to produce the smoothest possible images on the HoloLens. The available hardware and software consisted of the Optitrack system, a middleware (MotionHub) that can receive skeleton coming from a large pool of possible motion capture systems, a MotionHub plug-in for Unity that receives data from motionHub and animate an avatar on the Unity application and the HoloLens 1st Gen. AR headset. However, the MotionHub application had several limitations concerning this aim, and therefore had to be changed in order to meet the needs. So the first thing to start was to change the MotionHub application, and then to create an HoloLens application on Unity based on the MotionHub plug-in that could communicate with MotionHub, receive tracked skeleton and tracked HoloLens data and place the virtual character in the correct position. In the next sections, first, changes made to the MotionHub application and motivations will be described, then the HoloLens application and the choices made on it will be discussed.

5.2 MotionHub Changes

5.2.1 Introduction

MotionHub, as said, is an open-source middleware written in C++ that can receive simultaneously raw data coming from different BTS technologies, process it, creating skeletons with the same unified structure and transmit it to a third application client. However, MotionHub had some limitations that had to be overcome in order to implement this thesis system. First of all, MotionHub is made to receive only BTS data; the BTS data is the body tracking data, that means that it can only receive data of skeletons, and for this thesis also the HoloLens tracked data, which in Motive is seen as a single rigidbody had to be able to be received. So it had to be modified in order to receive, interpret and send the HoloLens position and rotation tracked by Motive to the HoloLens application. Secondly, although MotionHub uses NatNet SDK for communicating with Motive, it is unable to receive data if this data come from a different computer. So MotionHub has been modified in order to use its application on a different machine than the one with Motive installed on, and also by adding an interface to set the IP values of the machines. Finally, the last thing to modify was to add a button for sending an animation signal in order to be able to control the animation time of the virtual objects in the scene remotely while the actors are performing. In the following, changes made to the MotionHub open source application to meet the said needs will be presented.

5.2.2 Receiving Single RigidBody Data

The biggest change to make to the MotionHub application was to allow it to receive individual rigidbodies and not just skeletons, to allow it to process and also send the position and rotation data of the HoloLens tracked by the Optitrack system. But first, let's take a look on how MotionHub receives and interprets the skeleton data from Optitrack. The part of receiving data from the different BTS systems is delegated to various classes, each one referring to a particular BTS (for example for Optitrack there is the `OTTracker.cpp` class). Each of these classes is a derivative of the `Tracker.cpp` base class. This class has, as its attribute, a vector of skeletons: the `SkeletonPool`. The idea is each derivative class takes the data received from the BTS, transforms it into a standard skeleton object as defined by MotionHub and puts it in his own `skeletonPool`; then it will be the task of another class, the `NetworkManager.cpp`, to take this `skeletonPool` object, transform it into an OSC message and finally send it to the client (in our case the application). Going deeper into the code, the `Tracker.cpp` base class has a function: the `Update()` that runs continuously with a while loop and does two main things. The first one is to call the `track()` function which is

different for each derived class concerning a particular BTS, and the second one is to call the `NetworkManager.cpp` function `sendSkeletonPool()`. In our case the `track()` function called is the one implemented in the `OTTracker.cpp` class. This class receive data from Motive: the Optitrack software using the NatNet SDK, and so the structure of the data received depends on the NatNet SDK data types. In particular the NatNet SDK has two different types of packet to send for each frame: the Dataset Descriptions that contains descriptions of the motion capture data sets for which a frame of motion capture data will be generated (e.g. `sSkeletonDescription`, `sRigidBodyDescription`), and the Frame of Mocap Data that contains a single frame of motion capture data for all the dataset described from the Dataset Descriptions (e.g. `sSkeletonData`, `sRigidBodyData`) [81]. However, MotionHub to interpret the data uses only the Frame of Mocap data. The frame of mocap data contains for each type of data (e.g. `MarkerSet`, `RigidBodies`, `Skeleton` etc.) the list of the elements that are present in the current frame. So if a single skeleton is captured the list of skeletons will contain only one element but the list of rigidBodies will contain all the rigidBodies present in the skeleton and the list of the markers will contain all the markers present in all the rigidBodies of the skeleton captured. However in MotionHub, in the class `OTTracker.cpp` the function `extractSkeletonData()` is assigned to take the skeletonData present in the `FrameOfmocapData` that comes from Motive, and, for each skeleton, call the `parseSkeleton()` function that cycle on every rigidBody that forms the skeleton and create an object which is the MotionHub standardized version of skeleton. This skeleton is then added to the `skeletonPool` which is sent by the `NetworkManager.cpp` to the client. To send each skeleton the Network Manager calls another object the `OSCsender`. That is because MotionHub uses the OSC protocol to communicate with clients and so the data must be converted into an OSC message. The class `OSCsender` so calls its function `sendSkeleton()` that creates the OSC message which include as address of the message `/mh/skeleton`, then the unique id of the skeleton and then for each joint the position and rotation as 3D coordinates and quaternion values. So this is basically what happened in the MotionHub application.

In order to receive, interpret and send also the HoloLens position a lot of things had to be changed in this process. First of all, when receiving the data motionHub had to recognize the rigidBody that identifies the HoloLens. RigidBodies have two attributes for identification, first their id, and second, the name gave by user on Motive. The `FrameOfMocapData` only contains the list of the rigidBodies with their id but not the name of them and so it was not enough to understand if one of these was the one referring to the HoloLens. This is why the function `checkNewCameras()` was created. This function is called in function `track()` of the `OTTracker`, and uses the `DataSetDescription` data, received from Motive via NatNet, to check if there is a rigidBody with the name "HoloLens" (given on

Motive). If it finds it, the identifier of the corresponding rigidBody is saved in a CamerasID vector.

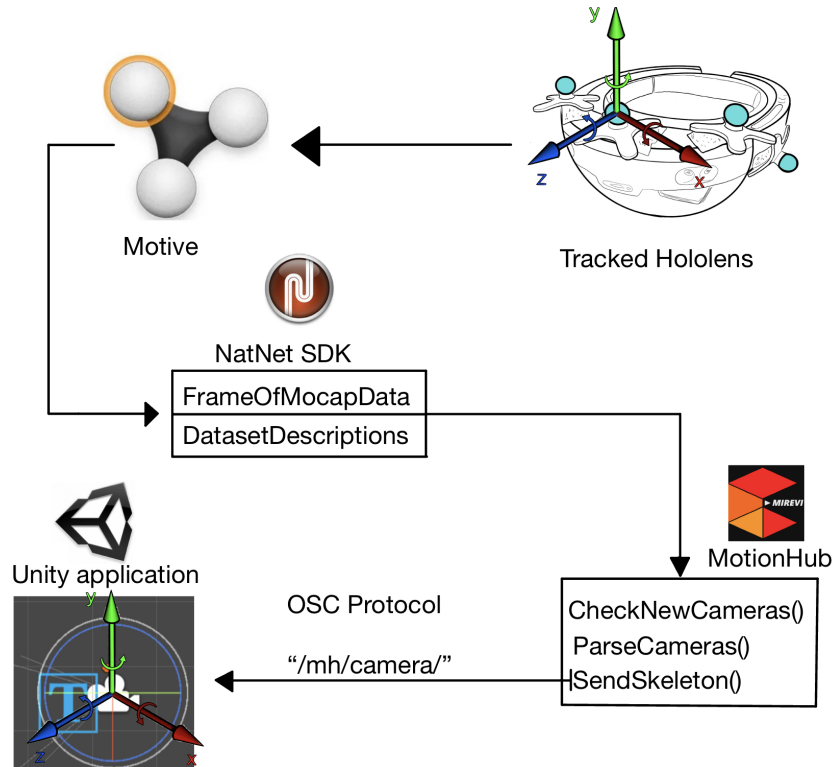


Figure 5.1: The path made by the tracked HoloLens position and rotation data from the OptiTrack system to the Unity application.

This control is made only when the number of rigidBodies (calculated using the data of the FrameOfMocapData) changes, in order to reduce the overall cost of the function. After that, the function `extractCameras()` checks if one of the rigidBodies received from Motive has the same id as one of those contained in the camerasID, and if so, calls the function `parseCamera()` which create a “fake” skeleton in which each joint has the same values as position and rotation which are the same of the rigidBody referring to the HoloLens. To distinguish between a skeleton and a fake skeleton which actually represents the rigidbody of the HoloLens, a boolean attribute is added to the skeleton object, named `isCamera`, which is set to true when creating a skeleton in the `parseCamera()` function. Finally, the function `sendSkeleton()` of the class `OSCsender.cpp` controls if a skeleton in the pool is actually a fake skeleton representing the rigidBody data of the HoloLens by checking the `isCamera` value of the skeleton object, and, if so, sends the data using the OSC protocol adding as address `/mh/camera`.

5.2.3 Communicating with Motive and Sending Signals

MotionHub communicates with the Optitrack software Motive using the NatNet SDK which integrates seamlessly with standard APIs (C/C++/.NET), and lets MotionHub communicate with Motive via a client/server network based on UDP. However, MotionHub, as said, was not able to communicate with Motive if that software was not installed on the same machine as MotionHub.

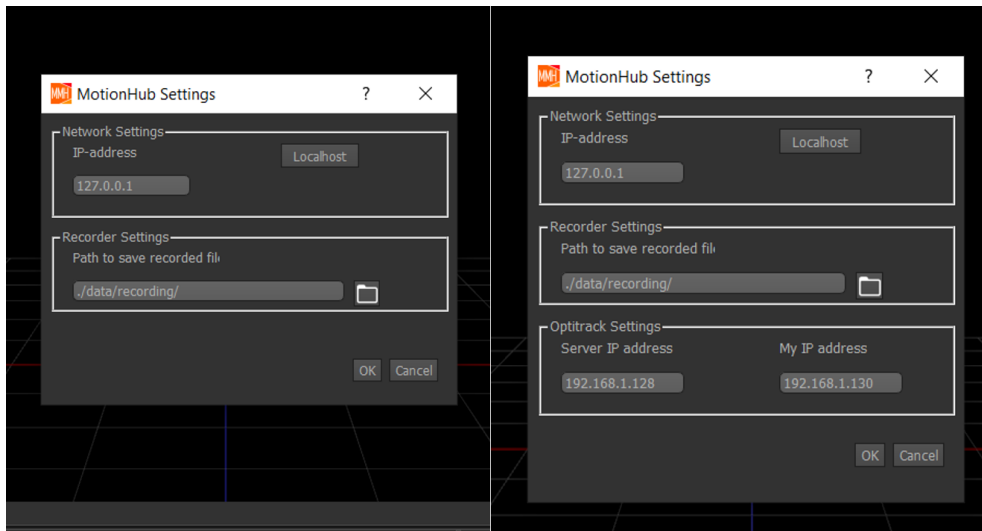


Figure 5.2: The MotionHub settings window before and after the interface modification for being able to connect with Motive from another machine.

So the MotionHub interface was changed (Fig. 5.2) in order to be able to set the IP address of the computer with MotionHub and the computer with Motive, then these IP addresses were used to create a NatNet client to receive data from Motive. The IP addresses are then saved on the configuration file of MotionHub, in this way, when reloading the application, it automatically reads the values from the files and, when the Optitrack tracker is enabled on MotionHub, tries to connect with the target machine to receive the Motive data of tracking. The HoloLens application has to be notified when an animation is started on a virtual object in the scene. This can be useful when rehearsing a scene in which there is not only the virtual character controlled by the actor through motion capture but also other virtual objects with which the actors have to interact or in case animation effects need to be added to virtual character itself.

To this aim, a button is added (Fig. 5.3) on the MotionHub interface that sends to the application an OSC message with no data but using `/mh/animationSignal` as the address. On the application side, an AnimationController was developed that, when this type of message is received, controls which animation to start.

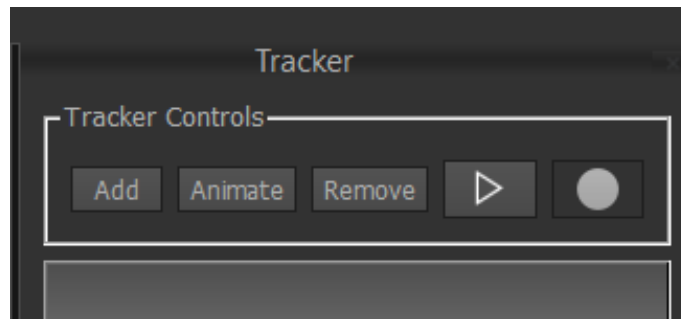


Figure 5.3: The animate button added on the MotionHub interface to send animation signals.

5.3 HoloLens Application

5.3.1 Introduction

The application uploaded on the HoloLens was created on the Unity game engine. This application first receives the data arriving from MotionHub, regarding the skeletons tracked by a BTS system (i.e., the Optitrack), and the position/rotation of the HoloLens traced by the same tracking system. Secondly it uses this data to allow an avatar, whose movements are piloted by the person traced by the BTS, to be displayed on the HoloLens in the correct position, i.e. in correspondence with the traced person himself. The application can also receive other types of signals from MotionHub which can trigger other events such as the one to launch animations. The application has also a mode called “Calibration mode” that lets the user adjust manually the position of the avatar by adding an offset in the x, y, and/or z axis. This application was developed by using the Mixed Reality Toolkit, which adds various tools to ease the development of HoloLens apps with Unity.

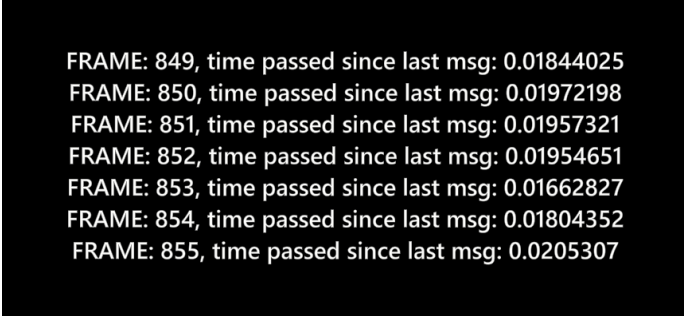
In the following, each part of the application will be described in detail.

5.3.2 Communication with MotionHub

The application is based on the unity package made by MotionHub which is an example implementation of a client that receives skeleton data from motionHub and animates an avatar on unity. To understand the changes done on this package to build the application, it is first presented how this package works. First of all, the connection with MotionHub is managed by the UDPPacketIO class that opens and closes the UDP connection with MotionHub, and receives the packet. UDP is preferable to TCP as it offers less latency. However the OSC class creates a new Task called `Read()` which continuously call the UDPPacketIO function `ReceivePacket()` that takes a data packet on incoming data. In the OSC class

the packet received they are decoded according to the OSC protocol and put into an ArrayList of messages. In the `Update()` which is a function that Unity calls every application frame, for each message in the ArrayList, the address of the message (as per the OSC protocol) is checked and a specific function to manage that type of message is called. For the skeleton messages the address is `/mh/skeleton` and the function called to handle it is `OnReceiveSkeleton()` in the `AvatarManager` class.

The `AvatarManager` takes care of all the avatars present in the scene, it creates them, destroys them and updates their position and pose when a skeleton message is received. In the `OnReceiveSkeleton()` function the message is decoded, and the position and rotation data of each of the 21 joints of the skeleton with the `skeletonID` equals to the one in the message, are used to update the skeleton position. A good thing about this Unity package is that it let developers animate the avatar as long as it has a humanoid rig on Unity. So this is basically how the `MotionHub` unity package works. In order to create the application, a few changes had to be done. First of all, to communicate via UDP this package used the `System.Net.Sockets` libraries which cannot be used by `HoloLens`. `HoloLens` must use the `Windows.Networking.Sockets` to communicate via UDP. This connection is managed by the class `UDPPacketIOUWP` that works in a parallel thread, takes the package coming from `MotionHub` and then sends it to the `OSC` class to be decoded. The IP addresses of the `HoloLens` the computer running `MotionHub` can be setted in a `.txt` file that is loaded on the `HoloLens`. Speaking about latency the packets are received from `motionHub` and decoded in ~ 0.019 s average as it can be seen in Fig. 5.4 giving more than 50fps on the motion captured avatar animation.



```
FRAME: 849, time passed since last msg: 0.01844025
FRAME: 850, time passed since last msg: 0.01972198
FRAME: 851, time passed since last msg: 0.01957321
FRAME: 852, time passed since last msg: 0.01954651
FRAME: 853, time passed since last msg: 0.01662827
FRAME: 854, time passed since last msg: 0.01804352
FRAME: 855, time passed since last msg: 0.0205307
```

Figure 5.4: Debug logs printed on the `HoloLens` every time a skeleton packet was received and processed.

These ~ 30 ms, that correspond to the latency time in average required by `Optitrack`, have to be added and thus ~ 50 ms is the total latency that occurs from the moment the performer with the motion capture moves to the final virtual avatar movement. This value can be considered acceptable.

5.3.3 Use of the Tracker's Data

So when received packet from MotionHub, the `UDPpacketIOUWP` calls a function of the OSC class: `ReadAndStoreMessages()`. This data acquisition happens in a separate thread. The two threads use concurrent stacks to pass each other the data as shown in Fig. 5.5. In fact the `ReadAndStoreMessages()` function, for each OSC packet, reads the address and put the OSC message in the correspondent Concurrent Stack. There are mainly four types of messages used in the application and so four corresponding concurrent stacks: Skeleton for avatars, Camera for the HoloLens position/rotation data, Animation for the animation signal and calibration that handles the message containing the position/rotation of the rigidBody that corresponds to an object used to have a reference of the real space for place virtual objects in the scene. So this is useful to put virtual objects in a specific real position.

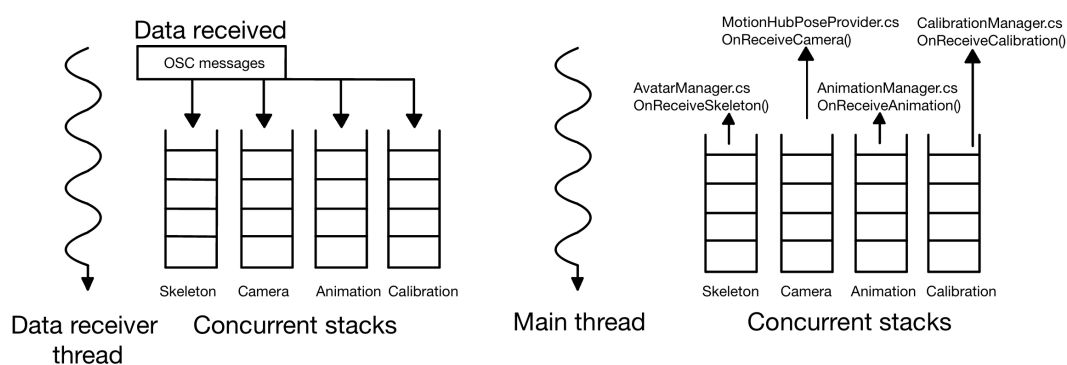


Figure 5.5: How the two threads of the application works to give/take and use the data received from MotionHub.

So, while the avatar data received are basically used as it was in the MotionHub unity package, so calling the `OnReceiveSkeleton()` function in the `AvatarManager` class, the camera data are treated differently. First of all, how Unity virtual camera works with HoloLens is discussed. In Unity the object that determines the point of view from which a virtual and watched scene is a virtual camera called the main camera. When using other devices like HoloLens a particular script called `TrackedPoseProvider` can be added to the camera object in order to provide the position and rotation of it based for example on the device position and rotation. For example, it is possible to create a Unity application to be runned on the HoloLens in which the virtual camera (the `mainCamera`) has a `TrackedPoseProvider` that takes position and rotation from the HoloLens IMU (inertial measurement unit). In this way, moving the HoloLens in real life would correspond to moving the virtual camera on the Unity application, allowing to navigate in the virtual space just by moving in the real one. However, when the HoloLens

application starts it takes the current Y rotation as the zero rotation and so it has its own coordinate system, which is different from the ones from the motion capture tracking system (Optitrack). So what is the best way in the Unity application to use the position and rotation data coming from the Optitrack system (passing through MotionHub) concerning the tracked HoloLens position and rotation to set the mainCamera in the correct position and rotation in the virtual world to see the avatars superimposed on the person wearing the mocap suit? The first approach was to use the tracked data directly to set the mainCamera position and rotation. This was done by changing the TrackedPoseProvider in order to give the mainCamera the position and rotation received from MotionHub. In particular when a message having the address /mh/camera arrived it was handled by the function `OnReceiveCamera()` in the TrackedPoseProvider that uses the data to set the mainCamera position and rotation. Fig. 5.6 shows how the camera data was used in the first approach.

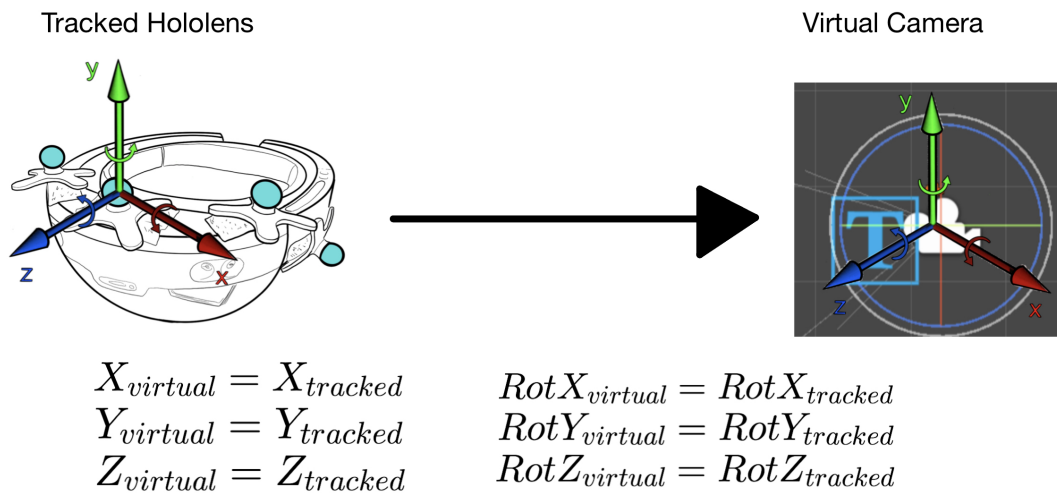


Figure 5.6: How the HoloLens tracked data was used in the first approach.

However, this approach proved to be unsuccessful. Even though the avatar was displayed in the correct position and covered the person with the motion capture suit in the right way, each real movement done by the HoloLens user, to affect the displayed image, had to pass through Optitrack, then MotionHub, and then arrive on the HoloLens, and change the mainCam position and rotation to change what the user see. And therefore it was not instantaneous. For example, if the user was looking a person with the motion capture suit through the HoloLens, and so he or she was seeing the avatar covering him, if he or she suddenly turned the head on the right, the avatar image on the HoloLens took a while to change according to the movements to show the virtual avatar from the correct point of view. This was a real problem especially for rotation because it made the application unusable

also as it could cause sickness. Apart from this, with this approach, in order to use the application, the user with the HoloLens must be tracked all the time and so must stay in the Optitrack tracked space. So using the position and rotation of the HoloLens as captured by the tracking system to place the virtual camera in the right position was not a good approach.

The second approach was to use the position and rotation data to control a second Unity camera that is called fakeCamera. The idea was to move mainCamera (which, as said, is the camera that determines what the user sees) using the HoloLens IMU, in this way avoiding all the possible sickness problems because the mainCamera position changes instantly as the HoloLens move, and, in the application change the position of all the virtual elements except for the fakeCamera so that the mainCamera, and so the HoloLens, would see as the fakeCamera (piloted by the HoloLens tracked data) was seeing. So when launching the application, you had to make sure that the tracked system was receiving the HoloLens tracking right and then do the airtap gesture (Fig. 5.7) to perform the calibration.

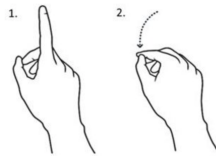


Figure 5.7: The AirTap gesture.

Performing a calibration means that all the virtual elements in the application scene, including the avatars and not including the fakeCamera, was moved so that the mainCamera (the user with the HoloLens) saw them as the fakeCamera (who was in the position/rotation of the HoloLens tracked) was seeing them before the calibration. This is explained in Fig. 5.8. This allows the user to leave the area traced by Optitrack since, after calibration, the HoloLens tracking data is no longer used by the application.

This method turns out to work in theory but it has a big disadvantage. The problem was that changing the avatar position in the virtual world could create some SRS problems if the calibration was not done perfectly. This happens when, while calibrating, the x and/or z rotation calculated by the HoloLens IMU are not exactly equal to the x and z rotation of the rigidBody corresponding to the HoloLens tracked by the Optitrack system. If this happens the horizontal line of the virtual avatar wont match the real one anymore, and so while the person wearing the motion capture suit for example walks on the ground normally, the 3D avatar will not follow his horizontal path but an inclined one leading to a drift problem. The more the person with the mocap suit walks the more the 3D avatar would be misaligned to him or her as it can be seen in Fig. 5.9.

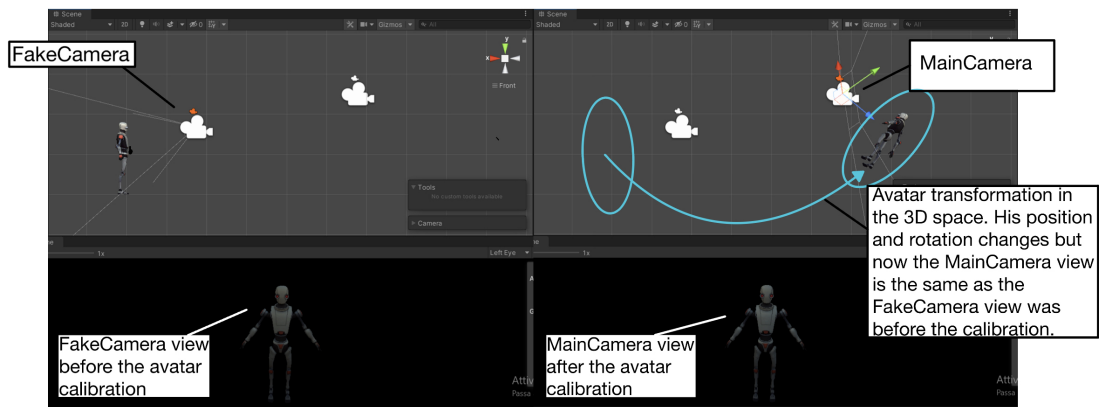


Figure 5.8: How the second approach works.

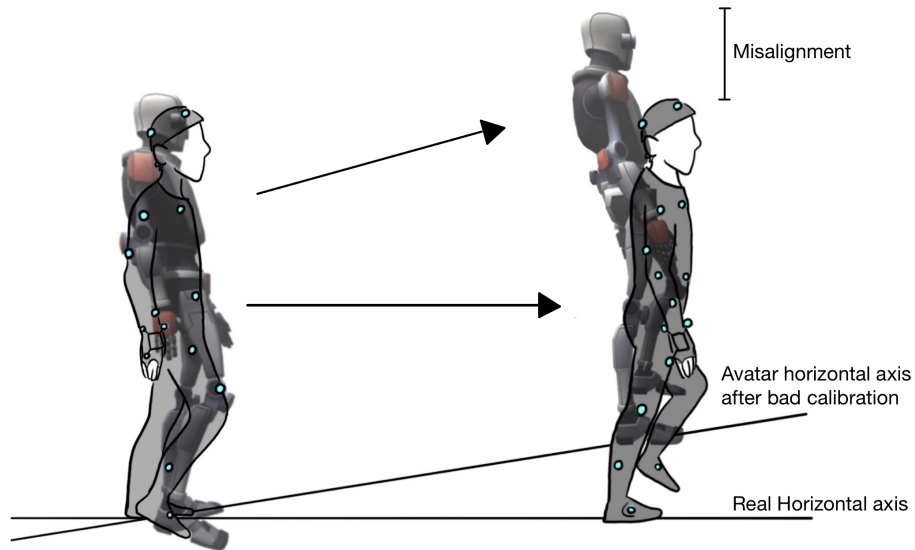
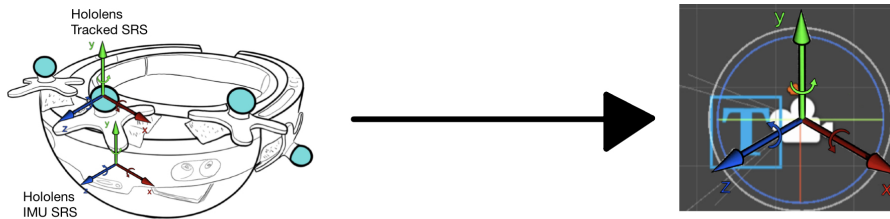


Figure 5.9: The second approach problem.

This is the reason why the third and successful approach was not based to change the virtual elements position. In this approach, at first, the HoloLens tracked position and rotation are used, but not entirely. In particular, by changing the TrackedPoseProvider script, the mainCamera takes the relative position of all three axes (x,y,z) and only the rotation relatives to the y from the tracked data. The rotation relatives to the x and z axis are taken from the IMU of the HoloLens. In this way, the virtual horizontal line would always correspond to the real one. However by doing so we would still have the problem seen in the first approach with regard to the position on the x,y and z axes and the rotation on the y axes. So the image shown on the HoloLens is not smooth and does not change instantly

when the user moves his head in the space or rotates it with respect to the y-axis. To overcome this issue I added a new type of calibration. The user put on his HoloLens with the markers on it, stays in the Optitrack tracking space, and when he or she perceives that the image is stable and the virtual avatar is "matching" the person in the motion capture suit he or she can do an AirTap to calibrate. In the few seconds later user must stand still to ensure that the HoloLens is tracked as well as possible and then, since when the calibrated message appears, the mainCamera and so the image seen on the HoloLens wont depend anymore on the HoloLens tracked position.



Before Calibration

$$\begin{aligned} X_{virtual} &= X_{tracked} & RotX_{virtual} &= RotX_{IMU} \\ Y_{virtual} &= Y_{tracked} & RotY_{virtual} &= RotY_{tracked} \\ Z_{virtual} &= Z_{tracked} & RotZ_{virtual} &= RotZ_{IMU} \end{aligned}$$

After Calibration

$$\begin{aligned} \Delta X_{IMU} &= X_{IMU} - prevX_{IMU} & X_{virtual} &= prevX_{virtual} + \Delta X_{IMU} & RotX_{virtual} &= RotX_{IMU} \\ \Delta Y_{IMU} &= Y_{IMU} - prevY_{IMU} & Y_{virtual} &= prevY_{virtual} + \Delta Y_{IMU} & RotY_{virtual} &= prevRotY_{virtual} + \Delta RotY_{IMU} \\ \Delta Z_{IMU} &= Z_{IMU} - prevZ_{IMU} & Z_{virtual} &= prevZ_{virtual} + \Delta Z_{IMU} & RotZ_{virtual} &= RotZ_{IMU} \\ \Delta RotY_{IMU} &= RotY_{IMU} - prevRotY_{IMU} & & & & \end{aligned}$$

Figure 5.10: How the HoloLens tracked data is actually used to place the virtual camera in the correct position and rotation.

From then on, the HoloLens position and y rotation are calculated in an incremental way only using the HoloLens IMU data. The application saves the x,y,z position and y rotation data acquired from the tracking system when calibrating and, from then on, on each frame it adds to that position the difference between the HoloLens position on x,y,z and rotation on y on the previous frame taken from the IMU, and the actual position on x,y,z and rotation on y taken from the IMU. So the tracked data is needed only at the beginning, before the calibration, and then the HoloLens is free also to exit the Optitrack tracked space always being able to see virtual objects in the correct position. Fig. 5.10 shows how the tracked data and the data coming from the HoloLens sensors was used to position the virtual camera correctly.

5.3.4 Application

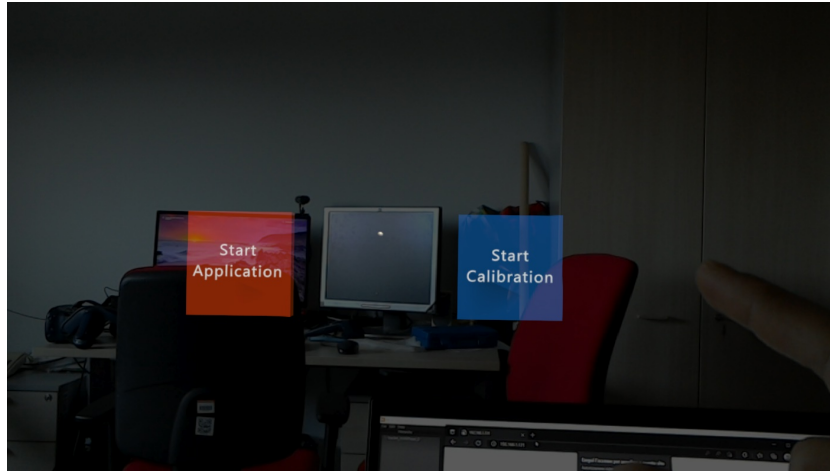


Figure 5.11: Menu.

As soon as the application starts, two modes are proposed to the user (Fig.5.11). These two modes are proposed in the form of two rectangular buttons, which are holograms of parallelepipeds, which the user can select by directing his or her gaze towards one of them and then making the airTap gesture. The proposed modes are the application mode and the calibration mode. The application mode is the standard mode in which data is received and, through the process explained in the previous sections, is used to show the 3D avatar over the mocap-suited actor. The calibration mode, instead, is useful for a manual calibration of the virtual avatar position in 3D space. In this mode apart from the 3D avatar a set of three sliders appears (Fig. 5.12).

The user can add a positive or negative offset on one of the axis by pointing his gaze to the corresponding plus or minus of an axis and do the airTap gesture. The avatar will move accordingly in relation to the set offsets. The offsets values are then saved in a file so that when reloading the application, when positioning the avatar using the tracked data received, it will automatically consider also the offsets chosen, and position it accordingly. This can be useful first when the avatar scale is bigger or smaller than the mocap actor scale. That is because regardless of the size scale of the virtual character, it will be positioned so that its root (i.e. the centre of its body, at the hips) coincides with that of the person piloting the avatar through the motion capture suit; and so for example in the case the avatar is bigger then the human scale, and so the avatars feet “enters” into the floor, one might want to place the virtual avatar higher augmenting the offset on the Y-axis. If the avatar is in a smaller size scale than the human one, one might want to set the avatar higher like to position it at the actor’s face, or lower, at actors leg. To

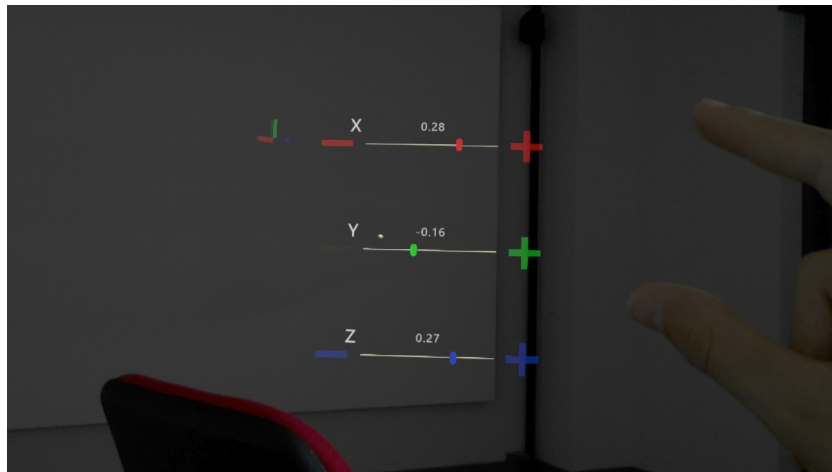


Figure 5.12: The sliders in the calibration mode.

return to the main menu the user can do two airTap gestures quickly while not pointing to any interactible hologram (like the “plus” or “minus”).

Chapter 6

Experiment

6.1 Introduction

The intention of the experiment is to show how augmented reality can be useful for actors when rehearsing or filming scenes in which motion capture is used. The next sections will discuss about how this experiment took shape, starting with other similar works used as a reference, then talking about the best use case study to be analysed, then the proposed experiment methodology and finally the experiment itself.

6.2 References

In order to design an experiment as good as possible, two works proposing similar experiments (already mentioned in Chapter 2) were used as references. These two papers presents work similar to the one made of this thesis, with related studies conducted as experiments involving actors and non-actors, from which to take cues as to how the experiment should be conducted and what aspects should be analysed.

6.2.1 First Reference

In [68], Kammerlander et al. have proposed using virtual reality headsets to help actors in scenes where motion capture is used to animate characters of different scales. By equipping both actors with an headset, they use virtual reality to allow them to see themselves as the virtual character they play and see the other from the perspective of their own virtual character (Fig. 6.1). Using virtual reality when filming this type of scene would produce several advantages:

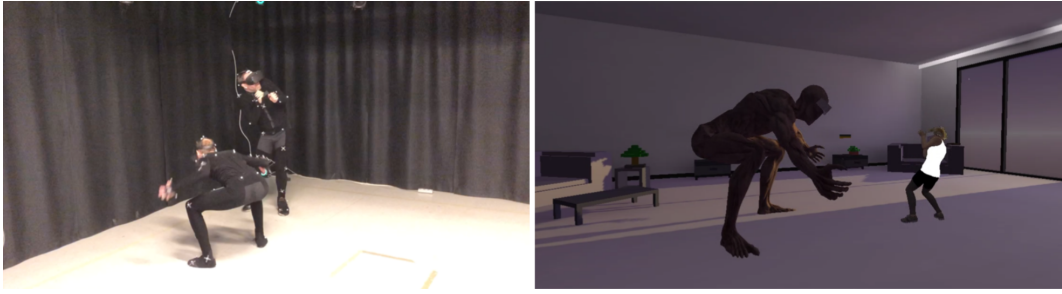


Figure 6.1: Actors shooting using VR for the experiment presented in [68].

- the actors gain a greater sense of “body ownership”, i.e. they better feel the body of the virtual character they are playing as if it were their own, which is useful especially when this character is of a different scale from the actor. So this allows the actors to identify more with the character they are playing improving the performance,
- by using visuals, the actors are immersed in the virtual scene and feel more mentally involved in the experience, which decreases the imaginative work they are usually forced to do when shooting these scenes, and
- it decreases the work done in post production by the animators. This is because generally, by shooting these scenes without having an actual conception of the size of their virtual character, the actors do not position themselves correctly (e.g., with respect to the virtual objects that will be present in the scene but that they do not see) or do not look the right way (e.g., if they have to look at a virtual character that is smaller or bigger than them) and it is the task of the animators in post production to manage these situations; whereas by shooting the scenes in VR, looking right from the point of view of their virtual characters, the actors manage to position themselves and direct their gaze more correctly.

However, using virtual reality also has a disadvantage. Wearing VR headsets, actors cannot actually look into each other’s eyes, but only see virtual characters, which can make it more difficult to get emotionally involved in the scene, to empathise with the other actor, which is can be crucial especially in more emotionally important scenes. To evaluate this system, a study was conducted whose objective was to compare the results obtained by acting in the traditional way and those obtained using virtual reality. A script was then created of a scene that the actors were to perform. The idea was to write a scene that would emphasise the advantages of using VR but that would not be too focused on the aspects to be emphasised. The scene to be filmed is that of a girl who finds herself in a room in her house in

tiny dimensions and encounters a huge monster, at first she is frightened then her interaction with the monster becomes more and more friendly and the two of them copy each other's movements, until she realizes she is in a dream and asks him how to get out of it; the monster opens a door and she comes out of the dream. The scene leaves enough freedom to the actors, describing only the general context of the scene but manages to show the advantages of using VR: 1) using two characters with totally different scales, 2) using a monster as a character, so it is important for the actor to identify with it and for the other to react as naturally as possible to its sight (which is helped by virtual reality), and 3) making the two characters copy their own movements, which, if the scene were shot in the traditional way, could generate a lot of positioning and animation problems. In this way, they show how VR can therefore lighten the work of animators in post production. The studio involved 22 people with different levels of acting experience. Two modes were planned.

- Motion capture only: in a first part a screen was used to project the virtual character moved live by the actor's movements, while in a second part the scene was filmed three times. Reference points in the studio were used to identify objects in the virtual world.
- Virtual reality and motion capture: in the first part the actors wore a headset and, in the virtual world, looked at their character in the mirror, in the second part they performed the scene three times.

The participants tried out both modes, some started with the first, others with the second, and then had to fill in a questionnaire and were then interviewed asking for opinions and impressions. The questionnaire was based on three metrics.

- Body ownership: to measure how much the actors felt the virtual body as their own. For this part of the questionnaire, five items were taken from [82] which provides a standard method for assessing the sense of embodiment in virtual environments.
- Social presence: to measure how much the actors felt each other's presence in the scene, which is important as it generates mutual understanding and greater coordination in acting. For this part of the questionnaire, two items were taken from [83] and three from [84].
- Mental immersion: to assess how mentally immersed and involved the actors felt in their collaborative acting experience. For this part of the questionnaire, two items were taken from [84] and one from [85].

The interview finally asked which acting setup provided better performances and why, whether virtual reality modified one's acting and if so how, and whether virtual reality influenced one's imagination and if so how.

6.2.2 Second Reference



Figure 6.2: The experiment done in [67].

In [67], Berthelot et al., as already mentioned in Chapter 2, propose a system to help actors when they have to shoot scenes with green or blue screens and have to interact with a virtual character. They propose to use virtual reality in the rehearsal of these types of scenes in order to help the actors practice their performance of the scene by making them see the virtual elements present. In the paper, they identify the main problems an actor has in performing these scenes. They classify these problems into three categories:

- **Positional problems:** when the actor must be in a specific position or make a particular gesture without having any visual clue (e.g. if he/she has to dodge an invisible object or hide behind an invisible wall or interact with virtual character hands). Usually to help actors in this they use marks on the floor or some objects (e.g. sticks with balls attached or laser) to be used as positional reference.
- **Timing problems:** these arise from the fact that VFX are almost always done in pre-production and therefore actors have to adapt their performances to the timing of the VFX (e.g. if an actor fights with a virtual partner he will have to synchronise his movements with those of the animations). Generally, to help the actors in this, timing sounds are used: they give information on when to perform a certain action.
- **Gaze direction problems:** these occur when the actor has to follow a moving virtual object or has to interact with a virtual character he cannot see. Typically, actors are helped by means of lasers or tennis balls on sticks or cardboards to which the actor refers. In the paper, they present two systems: one in which the actor is immersed in the virtual environment of the scene and there are various props with which he or she can interact in order to become familiar with the environment, and the second, more complex system in which the actor can rehearse the scene with gradually increasing difficulty: initially the timing is dictated by the actor who is rehearsing, then it is the actor who has to rehearse within the timeframe dictated by the virtual environment.

They propose three ways of use:

- with large screens, 3D glasses and flystick to interact;
- with a projector, 3D glasses and flystick ;
- with an augmented reality headset and a controller to interact.

In order to evaluate this system, a study was conducted, the aim of which was to compare the results obtained from acting rehearsals conducted in the traditional way and those obtained through virtual reality acting rehearsals. Fig. 6.2 shows the three phases of the experiment. Two groups of 12 people each were taken, both with varying levels of acting knowledge (from inexperienced to experienced actors); the first was offered the traditional rehearsal system, the second the virtual reality system. Each member rehearsed the scene three times with the method proposed to him or her, then the scene was filmed and afterwards a questionnaire was proposed to them. Then they were requested to try the other method and finally they did a final mode preference questionnaire. The proposed scene was done in such a way as to be able to analyze certain aspects useful in evaluating the actors' performance. This scene saw the actor having to interact with a virtual tiger. Three "synchronization points" were defined in the scene, i.e., moments in which the actor interacts with the tiger and data can be derived.

- SP1: The tiger passes between the actor's legs. In this case, the actor has to pretend that a tiger passes between his legs at a specific time and in a specific position (timing and positional problem).
- SP2: The tiger is sitting at the bar and the actor is staring at it (gaze direction problem).
- SP3: The actor receives an orange from the tiger, it has to happen at a specific time (gaze direction and timing problem).

In the traditional method, they used sounds to give the timing, marks made with scotch tape to give the positioning, sticks with balls on them: one to define the position of the tiger, useful for directing the gaze, and another for the virtual orange. The objective data collected from the performances that were then to be compared concerned how much the actor was positioned in the correct position (distance between where he was and where he should have been), how much the actor was looking in the right direction (distance between where he was looking and where he should have been looking) and how much the actor was synchronized with time. For example, in SP1 the distance between the actor and the marker on the ground was calculated, in SP2 the distance between the point of intersection of the ray from the actor's eyes to the wall where the marker defining the tiger's

eyes is placed and the marker itself, and in SP3 the distance between the actor's hand and the position where the virtual orange should have been. The proposed questionnaire, after an identification section, in which there were personal questions, questions on knowledge of virtual reality and the acting experience, first proposed technical questions on the scene just rehearsed (e.g. where did the tiger come from?) and then a series of questions to which the user answered on a scale of 1 to 7 that sought to understand how much the rehearsal environment had helped the user:

1. in positioning;
2. in following the tiger better;
3. in feeling comfortable with their gestures;
4. to work on their facial expressions;
5. in working on their body expressions;
6. in getting emotionally involved in the scene;
7. in making better use of space;
8. in being more confident during filming.

Finally, after the users had also tried the other mode, there was a sequence of questions regarding the comparison between the two modes; users were asked which mode they preferred and why, and which mode is preferable for working on positioning, gaze direction and timing.

6.3 Use Cases

Now will be analyzed which use case should better be used for the experiment and, referring to the studies conducted in the papers just mentioned, which metrics can be evaluated depending on the use case. The scene will involve two actors, one playing a real character, and another playing a virtual one and thus using a motion capture suit, in a real context. With respect to this scene, there are mainly two use cases that can be analyzed.

- UC1: During the rehearsal of the scene, the actor playing the real character wears an AR headset that allows him/her to see the virtual character (superimposed on the actor wearing the motion capture suit) and the surrounding real environment.

- UC2: During the rehearsal and also in the shooting of the scene, the actor playing the virtual character and thus wearing the motion capture suit, uses the AR headset to see the virtual character's body parts superimposed on his own, but also additional parts that are part of the virtual character such as wings or arm extensions, etc.

The two use cases are illustrated in Fig. 6.3.

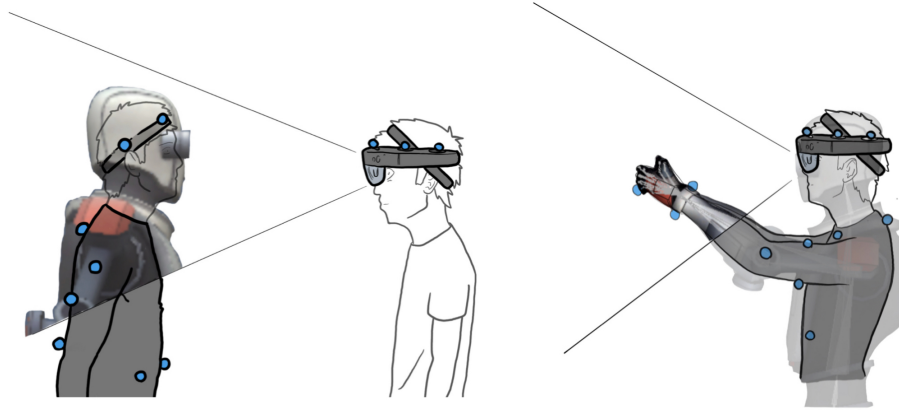


Figure 6.3: On the left an illustration of the UC1, on the right an illustration of the UC2.

Depending on the use case being considered, however, the aspects to be assessed and thus the type of test to be performed change. In both cases, the aspects described in [67] can be assessed, which in the case of the present work may concern how much the use of AR may have influenced one's facial and bodily expressiveness and emotional involvement with the scene. Other metrics can also be considered in UC1. First of all, as in [67], how much, in our case augmented reality, has helped the actor in the direction of the gaze (if the character has to look at a specific part of the virtual character's body). Spatial positioning can also be assessed, as done in [67], by, for example, tracking the hand of the actor wearing the headset. Then, as in [68] the social presence can be evaluated, i.e. how much the actor feels the presence of the other (the virtual character) in the scene and how natural it is to interact with it. In UC2, instead, other different aspects can be evaluated. First of all, the fundamental aspect is the one described in [68] concerning body ownership, i.e. how much looking at one's own limbs or other extensions of the virtual body one is playing, makes the actor feel as if that virtual body is really his own and how much this improves his acting performance; secondly, in UC2 the aspect of positioning can be better evaluated, i.e. when, seeing one's own virtual body extensions can help the actor to interact with them. Secondly, in this UC2 the aspect of positioning can be better assessed, i.e. when seeing one's own virtual

body extensions can help the actor to understand the real size of one's own body and thus better position oneself in the scene (e.g. avoiding that one's virtual body intersects real objects). In both cases, other aspects concerning more specifically AR should then be evaluated.

So, to summarise, the following aspects can be assessed in UC1:

- social presence;
- direction of gaze;
- spatial positioning (tracking the hand for example);
- emotional involvement, help in performing the scene (how much does it affect one's facial expressiveness and body expressiveness);
- user experience with respect to augmented reality.

In UC2, in turn, the following aspects can be assessed:

- body ownership;
- spatial positioning;
- emotional involvement, help in performing the scene (how much does it affect one's facial expressiveness and body expressiveness);
- user experience with respect to augmented reality.

Although in theory, both use cases would be valid, in particular, UC2 since it can also be used during the filming of the scene, there are some considerations to be made. First of all, while in UC2, in order to foster a sense of body ownership, it would be preferable to use virtual characters of the same scale as the actor, as in this way the actor's point of view would coincide with that of the virtual character. In this case, therefore, in order to show the advantages of the use of augmented reality, it would be necessary to use a character of the same scale as the actor as mentioned above, but with extra virtual parts such as arm extensions or wings, etc.; in UC1, on the other hand, a character on a different scale (larger or smaller) than the human one would be ideal in order to highlight how AR can help in directing the gaze (when for example the actor has to look the virtual character in the eyes, or has to see characteristic parts only of the virtual character). It is important to consider that this condition is the one that creates the greatest difficulties for actors, as they are forced to imagine a character more or less the size of the actor they see in front of them and with whom they interact (generally, if possible, props are used, such as cardboard boxes or padded overalls or stilts to make the actor's physiognomy similar to that of the virtual character he or she is playing).

Secondly, the technological limitation of the nowadays OST-HMD concerning the size of the FoV, works to the disadvantage of the UC2. In fact, in this case the main analyzable elements would be, as mentioned, body ownership and positioning. Having a small FoV, however, for the actor wearing the motion capture suit and headset and playing the virtual character, results in being able to see his or her own body as virtual only under certain conditions, i.e. only when, for example, his own limbs or the additional extensions of the virtual character enter that FoV, i.e. when he looks almost exactly there. And this prevents the advantages that the use of AR could give in terms of body ownership and positioning. It may be more difficult for the actor to “feel” the virtual character, being able to see only a small part of himself with the same part of the virtual character superimposed and only when he directs his gaze towards it. The same applies to positioning, for which the advantages of using AR would certainly be limited by the small size of the FoV. UC1 obviously also suffers from this technological limitation, however it is less affected by it and in particular only when the user is close to the virtual character (which in that case he/she would not be able to see completely); nevertheless, it still allows the various aspects described above to be assessed.

For these reasons, the experiment will focus on UC1: two actors, one playing a real character, and another playing a virtual one using a motion capture suit rehearse a scene letting the actor playing the real character wear an AR headset that allows him/her to see the virtual character (superimposed on the actor wearing the motion capture suit) other virtual elements in the scene and the surrounding real environment.

6.4 Experiment Procedure

Now that the use case has been identified, it is necessary to define the procedure of the experiment. The procedure is inspired by the one used in [67]. The idea is to compare two different methods of rehearsing a scene in which there are virtual elements (including a character animated live via motion capture): the traditional method (which uses props) and the proposed method (which uses an AR headset to make the actor visualize the virtual elements) and to see how much one or the other helps then in shooting the scene without the use of props or the headset by comparing subjective and objective parameters. The experiment then takes place in this way: the volunteer rehearses a scene with another actor, who wears the motion capture suit, three times using one of the two proposed methods. After which the scene is shot. When the scene is shot there are no scene props or headset to see the virtual elements; this to assess how much having experienced the scene in a certain method helped the actor in terms of directing the gaze, positioning, but also the emotional involvement in the scene. After which the volunteer will

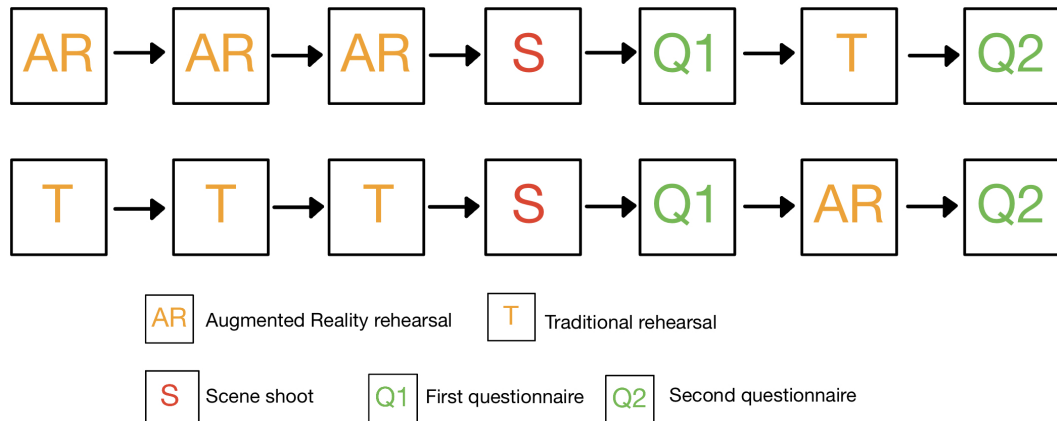


Figure 6.4: The experiment phases.

answer a questionnaire regarding:

1. the usability of the test system used;
2. the spatial presence or how much he felt part of the place where the scene was set when he tested it;
3. the social presence or how much he felt he could interact with the virtual characters during the rehearsal;
4. how much this rehearsal system helped him in actually shooting the scene with respect to various metrics such as the direction of the gaze, positioning, emotional involvement, etc.

Subsequently, the volunteer will also try the other test method that he has not tried before and will answer the first three questions of the previous questionnaire, plus a series of questions about his preferences on the test system. In addition, after testing the AR system, the volunteer will answer a questionnaire about their experience with augmented reality. The experiment procedure is summarized in Fig. 6.4.

6.5 Scene

6.5.1 Requirements

Since the intention of the experiment is to show whether, and in what way, AR can be useful to an actor in rehearsing scenes in which he/she acts with another actor playing a virtual character through motion capture, the scene to be proposed should

involve all the various difficulties an actor may find in having to rehearse and shoot such scenes, and it should also be clear that AR cannot be easily replaced by virtual reality for scene rehearsal. Difficulties can be summarised as follow: difficulty in directing the gaze, difficulty in positioning oneself and difficulty in having to react emotionally to events involving virtual objects/characters that the actor does not therefore see. The scene must therefore have the following characteristics.

- It must be emphasized that this technology helps in directing the actors' gazes when they have to look at virtual elements, which may mean that the virtual character animated in motion capture will have to be of a different scale from the human one (smaller or larger) or that the actor must look at a virtual object at a certain point.
- There shall be interactions between the actor and objects in the virtual world.
- Emotional reactions shall be required from the actor with respect to events involving virtual objects/characters.
- There shall be animations that are difficult to realize with physical props as in the traditional method.

Furthermore, since an AR system is proposed, it is evident that this cannot be replaced in rehearsing the scene by virtual reality. In particular:

- there must be real objects that will then actually be filmed in the scene with which the actors have to interact;
- actors must have to interact with these real objects in ways that are difficult to do in virtual reality.

6.5.2 Script

In order to create a script for a scene that fulfilled all requirements, several scenes from recent films in which motion capture was used were analyzed. Two scenes in particular were used to create this script. The first is from the film "The One And Only Ivan" (2020). In this scene, a little girl shows a gorilla (made in CGI and animated through motion capture) some drawings. From this scene, inspiration was drawn on using sheets of paper as real objects with which to make the virtual character interact, as actions such as drawing on them or tearing sheets of paper/crumpling them are actions that are difficult to reproduce in VR. But the real source of inspiration for making this script is the live action film "Aladdin" (2019). In this film, the character of the Genie was animated by actor Will Smith using motion capture. In particular, the scene that inspired us most is the one in which the Genie sings the song for Aladdin "Friend Like Me". A frame taken



Figure 6.5: A frame from the scene “Friend like me” from the movie “Aladdin”(2019) [86].

from this scene is shown in Fig. 6.5. Although this is a very complex scene with several CGI elements, it offered us several interesting insights such as the many interactions with even real objects and the fact that the Genie changes the appearance of his body several times (enlarging it at will and sometimes modifying it). The proposed scene is this: there are two characters: Aladdin and the Genie of the lamp, one (Aladdin) to be played by the volunteer participating in the experiment and the Genie played by an actor through the use of motion capture. In the scene, the character of Aladdin will have to help the Genie to free himself from a cage and, to do so, he will have to interact with him, and with other real and virtual objects/characters. The script is reported in the following.

INT. ALADDIN IS ON A MISSION TO RELEASE THE GENIE OF THE LAMP AND, AFTER HAVING CROSSED A THOUSAND PERIPECTIES IN THE CAVE OF WONDERS, FINALLY FINDS IN THE ROOM WHERE THE IMPRISONED GENIE IS. ALADDIN WILL HELP THE GENIE RELEASE FROM THE CAGE WHERE HE IS KEPT.

Aladdin enters a room and sees the Genie imprisoned in a cell in the distance in front of him. He hears the sound of a beast’s screams, turns to where the sound is coming from and is frightened at the sight of a beast that looks like a hell dog.

ALADDIN
(Looking at the dog) Aaaaa (Scared!
Taking a leap backwards)

The dog is hooked by a rope to a pole that presides over the cage with the Genie inside.

GENIE

Who is there??

ALADDIN

Genie I am Aladdin I came to free
yourself

GENIE

Ah it's you boy! Thank you. I
couldn't stand being locked up in
here anymore

ALADDIN

How do I free you?

GENIE

Should my spell book be there next to
you, could you tear off page 42 and
throw it at me?

I don't quite remember how to do the spell to get out of here. Aladdin agrees, finds the Genie spell book, tears off the page requested by the Genie, rolls it up and throws it at him. The Genie takes it and stops for a moment.

GENIE

Ah, one more thing, you should help
me get rid of this dog somehow

ALADDIN

Yes, but how?

GENIE

So I have a bone here, I could throw
it away to distract him, but we
should untie him from that pole first.
Do you have a way to cut the rope?

Aladdin thinks about it, then he remembers he has a knife with him.

ALADDIN

Yes!

GENIE

Then I throw the bone and you cut the
rope, ok?

ALADDIN

Agree!

Aladdin cuts the rope, but the Genie is still looking for the bone.
The dog growls and slowly begins to approach Aladdin.

ALADDIN

(frightened, staring at the dog
slowly approaching him)Um ... Genie
... the bone ??

Aladdin backs away frightened by the dog still staring at him in
fear. The Genie continues to search for the bone.

ALADDIN

(increasingly terrified from the
threat of the dog)Genie????

GENIE

There it is!!

The Genie throws the bone through the bars as far as possible. The dog starts running fast towards the bone thrown by the Genie. Aladdin follows the dog with his eyes and sees him go away.

ALADDIN

Fiuuu ... (relieved)

The Genie opens the sheet of paper to recite the spell.

GENIE

Ahh that's how it was! "Reduco mea
parti!"

A whirlwind of magic envelops the Genie and is transformed by shrinking into a mini Genie. Aladdin stares the Genie in the eye as he shrinks.

ALADDIN

Wooo (Amazed)

Aladdin approaches the Genie cage. The Genie then, now shrunken, comes out of the cage floating in the air. Aladdin walks over to the mini-Genie to see him up close and looks him in the eye.

GENIE

Thanks Al, since you were like that
kind to free me I want offer you one
of these two gifts in addition to
your three wishes

The Genie points his right hand towards the ground. Immediately after, a diamond appears which begins to whirl and finally stops above the Genie. Aladdin looks after the diamond as it spins in the air.

GENIE
You can choose from this diamond red
that will give you skills and
knowledge

The Genie points his left hand towards the ground this time. And a gem appears which, like the diamond, comes to life by whirling in the air. Aladdin looks after this too.

GENIE
Or this blue gem that will bring you
luck in life. Choose calmly then
write what you have chosen on the
last page of my book, and the gift
will be yours!

Aladdin thinks about it, then makes his decision and chooses the Gem of Fortune. Then he opens the book, goes to the last page, writes "Gem of Fortune", then closes the book of Genie. Now he can take the gem. He puts his hand close to the gem and takes it.

In the following, the script is commented in order to highlight how it meets the requirements discussed in the previous section.

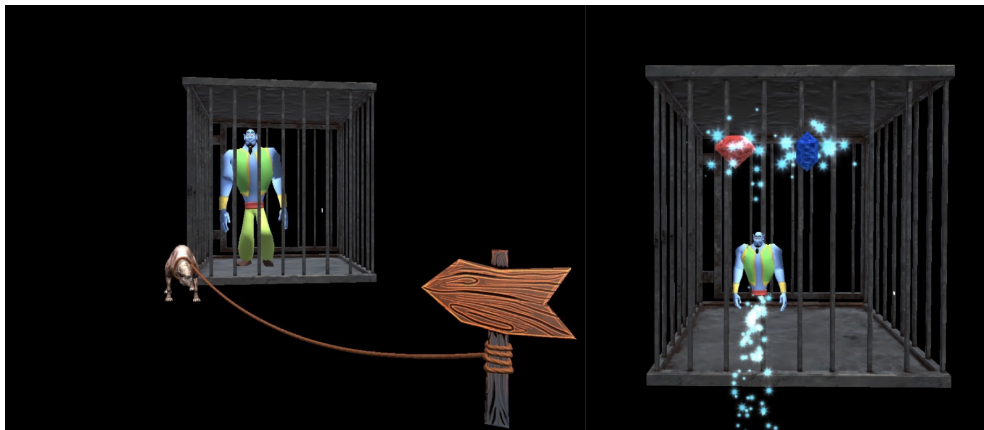


Figure 6.6: Two scenarios (at the beginning, and at the end) of the proposed script.

First of all, the Genie is on a different scale size than the human one: at first it is 2.10m tall then he downsizes to 50cm. This aspect provides the possibility to analyze the problem of directing the gaze concerning the interaction of an actor with a virtual character played by another actor who animates him through motion capture in both cases: 1) the character is larger than the actor who plays him and 2) the character is smaller than him. Speaking about eye-gaze the actor in many times in the script has to look exactly at a position defined by the virtual objects/characters. In particular when the dog is coming to him and he's terrified, when the Genie transform and the actor playing Aladdin has to look in his eyes while he's doing so, and when the diamond and the gem appears from the ground moving in a spiral. The actor playing Aladdin has also to interact many times with virtual and real objects. The virtual objects are the rope that he has to cut (using a real object, the knife) or the gem which he takes at the end of the script. This can be use to evaluate the positioning problem, in particular when he has to take the gem putting his hand on it. The real objects are the knife and the book. In particular, the character of Aladdin leafs through the spell book, tears a page from it, crumples it, throws it, and finally writes in the spell book, all actions are difficult to reproduce in VR.

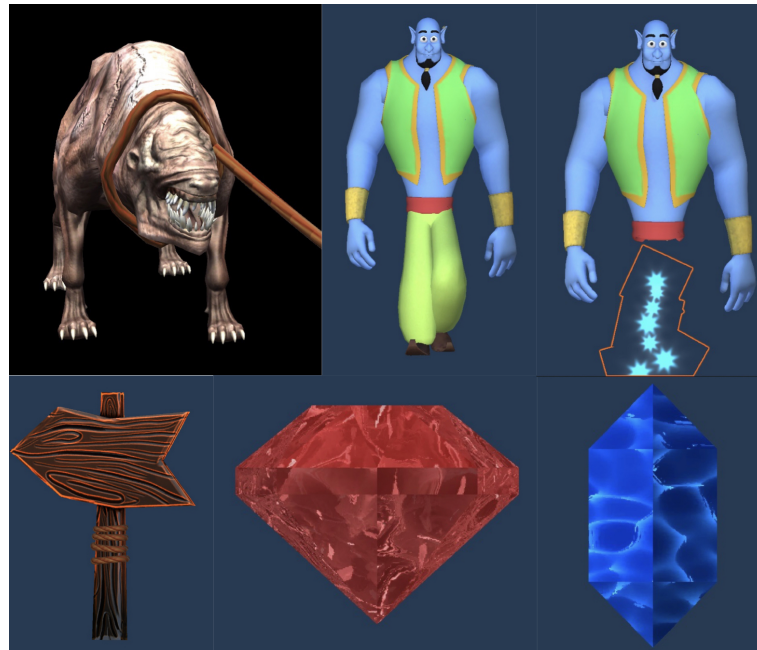


Figure 6.7: The virtual elements present in the experiment script scene.

Moreover, this scene presents two events in which the character has emotional reactions to events involving the virtual characters. The former is when Aladdin watches the infernal dog come threateningly toward him and the actor must pretend

he is terrified. The latter is when the Genie does the transformation and Aladdin looks at him amazed. Finally, the scene presents some virtual elements and movements that are difficult to reproduce using real props, like when the diamond or the gem appears.

6.6 Realization

In the following, details about the experiment will be presented. First of all, the volunteers were presented the meaning of the experiment and what they were going to do. After that, the script that would be tried and performed was analyzed together and the volunteers were shown images regarding the digital elements present in the scene itself (cage, Genie, dog, rope, diamond gem) and the real objects that they would have used (book and knife). The first section of the questionnaire, concerning general information on the participant such as age, gender, acting experiences, and knowledge of AR was then completed.

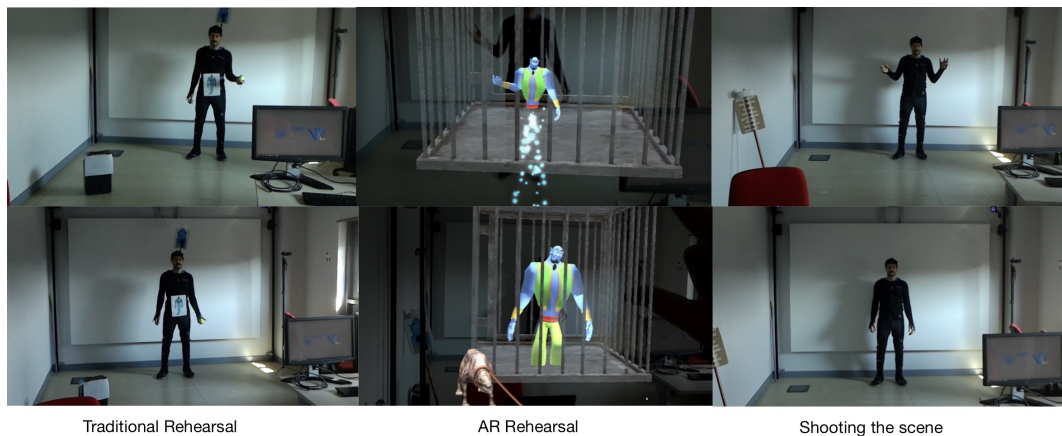


Figure 6.8: The volunteer’s view during the rehearsal with the traditional method (left), during the test with the AR method (centre) and during the shot (right).

After that, the scene was rehearsed three times either with the traditional method or with the proposed method or using the AR headset, depending on the group the volunteer belongs to. Fig. 6.8 the volunteer’s view during the two types of scene rehearsal and during the shooting.

The traditional method involved the use of these props:

- a rod with a hardcover of the Genie face on it to give a reference point on the direction of the gaze when the Genie is in its normal form;
- a cardboard figure depicting the Genie in its mini form, placed on the actor’s

belly with the motion capture suit always to give a reference on the direction of the look when the actor had to interact with the mini-Genie.

- a prop to indicate the presence of the dog in the scene;
- a laser to indicate the dog that is moving, in order to help the actor imagine something coming towards them and give him a clue as to the direction to look;
- a tennis ball, used in two circumstances: firstly when the Genie transforms to give a reference on the position of his eyes during the transformation and then when the diamond and the gem appear with their spiral movement.



Figure 6.9: From the left: 1) the tracked gloves used to track the tester hand, 2) the “magic book” and the knife prop used in the scene, 3) and 4) two props used to help the testers direct their gaze when interacting with the Genie (and Mini-Genie) during traditional rehearsal, and 5) a prop used to represent the dog and a laser pointer to give a reference on where to look to the tester when the dog was moving, during traditional rehearsal.

During the AR rehearsal, the volunteer used the HoloLens to see all the virtual elements in the scene (including the Genie, who was animated in real time using motion capture, and was superimposed on the actor piloting him through the mocap suit). In this methodology, the right hand of the user was tracked using a glove with markers in order to let him or her interact with the gem at the end of the script. In this way, the actor could actually take the gem in his or her own hand. After rehearsing the scene three times with the traditional or AR method the scene was shot. During the shot not a single prop was used, and the volunteer must use his/her imagination and memories of the rehearsals to play the scene as better as he/she can in terms of gaze-directioning, spatial positioning and emotional reactions. However, there were marks on the ground to indicate the position of virtual elements such as the sign, the dog and the cage. When shooting the volunteers playing Aladdin wore the HoloLens and the tracked gloves

in order to capture data concerning the eye gaze and the hand position. After the shooting, the volunteer had to answer a questionnaire regarding the tests of the scene in terms of spatial presence and social presence usability, and how much the test methodology of the scene had helped him then in shooting it. Finally, the tester tried the other rehearsal method that he/she haven't tried yet and answered the same questionnaire as before but concerning the other rehearsal method minus the part he had filled out earlier which was about how much the test methodology of the scene had helped him then in shooting it. Plus he/she had to answer to a series of questions concerning the AR experience in general and his/her rehearsal method preferences.

6.6.1 Questionnaire

The questionnaire as mentioned was made up of various parts.

- Pre-experience questions
- Post three times rehearsal questions
 1. Usability of the rehearsal system
 2. Spatial presence
 3. Social presence
- Post shoot questions (how much the rehearsal method has helped during shoot):
- Post both rehearsal methods questions (comparing the two methods)
- Augmented reality experience

6.6.2 Pre-experience Questions

The pre-experiment questionnaire in [67] were used, encompassing the following items.

1. Progressive ID
2. Age
3. Gender
4. Do you have any knowledge of theatre (acting)? 1 (none), 2 (minimal knowledge), 3 (some knowledge), 4 (good knowledge), 5 (excellent knowledge)

5. Have you ever acted? 1 (never), 2 (may have happened), 3 (occasionally), 4 (quite often), 5 (every day)
6. When was your last acting experience?
7. Have you ever acted in scenes where motion capture was used? 1 (never), 2 (may have happened), 3 (occasionally), 4 (quite often), 5 (every day)
8. What knowledge do you have about augmented reality? 1 (none), 2 (minimal knowledge), 3 (some knowledge), 4 (good knowledge), 5 (excellent knowledge)
9. How often do you use augmented reality systems? 1 (never), 2 (may happen), 3 (occasionally), 4 (quite often), 5 (every day)

6.6.3 Post Three Times Rehearsal Questions

Usability of the Rehearsal System

For these questions, [87] was used, asking to give a score to the following statements. 1 (Completely disagree), 5 (Completely agree).

1. I guess I could use the scene testing system (scene props or AR headset) frequently.
2. I found the scene testing system overly complex.
3. I think the scene test system is easy to use.
4. I think I would need the support of a technician to understand how to use the scene test system.
5. I found the various features of the scene testing system well implemented.
6. I think there were too many inconsistencies in the scene testing system.
7. I think most people would learn to use a scene testing system like this quickly.
8. I found the scene rehearsal system inconvenient to use.
9. I felt confident/safe using the scene rehearsal system.
10. I needed to learn a lot of things before I could use the scene test system.

Spatial Presence

These questions were taken from [84] and had to be answered on a scale from 1 (not at all) to 7 (very much).

1. How much did you feel that the objects you saw/heard/imagined were part of the environment you were in?
2. How far did you feel you could reach out and touch the objects you saw/heard/imagined (such as the rope or gemstone)?
3. How many times when you had to see/imagine an object coming towards you (like the dog walking towards you) did you instinctively move?
4. To what extent did you experience the feeling of being there: within the environment you saw/heard/imagined?
5. To what extent did the sounds seem to come from the objects you saw/imagined?
6. How many times have you had the instinct to touch or pretend to touch objects you were seeing/imagining even though it was not explicitly stated in the script?

Social Presence

Also these questions about social presence were taken and modified from [84], and had to be answered on a scale from 1 (not at all) to 7 (very much).

1. How many times did you feel that the characters (Genie and dog) you saw/heard/imagined in the scene could also see/hear you?
2. To what extent did you feel you could interact with the characters (Genie and dog) you saw/heard/imagined?
3. How much did you feel that, when required by the script, your movements depended on those of the other characters (Genie and dog) (e.g. when you moved in response to the dog's approach)
4. How much did it seem that you and the characters (Genie and dog) you were seeing/hearing/imagining were in the same place?
5. How often did you feel like the Genius was speaking directly to you?
6. How many times have you wanted to or made eye contact with one of the characters (Genie and dog) that you have seen/heard/imagined?
7. How protagonist-like did you feel about the interaction with the characters (Genie and dog), which you saw/heard/imagined?

6.6.4 Post Shoot Questions

These questions were inspired by the those used in [67], and had to be rated on a scale from 1 (completely disagree) to 7 (completely agree).

1. The rehearsal system helped you position yourself better in the space during the shoot (e.g. when you had to cut the rope, approach the Genie or pick up the gem).
2. The rehearsal system helped you follow the Genie better during the shoot (e.g. when he was in mini-Genius form and walking towards you).
3. The rehearsal system helped you feel more comfortable with your gestures during the shoot (e.g. throwing the paper or cutting the rope).
4. The rehearsal system helped you better manifest the emotional states of the character you were playing through your facial expressions (e.g. when you had to feign fear for the dog or astonishment at the Genie's transformation).
5. The rehearsal system helped you better manifest the emotional states of the character you were playing through your gestures (e.g. when you had to feign fear for the dog).
6. The rehearsal system helped you work on your emotional involvement in the scene in general.
7. The rehearsal system helped you make better use of the space (e.g. in figuring out where the dog was or approaching the mini-genius).
8. The rehearsal system allowed you to be more confident in your performance during the shoot.
9. How ready did you feel to shoot the scene? (1 Not at all ready, 7 Perfectly ready)

6.6.5 Post Both Rehearsal Methods questions

Also these questions were inspired by those in the questionnaire used in [67], and asked to choose between AR or Traditional method.

1. In your opinion, which shooting test mode is preferable for working on positioning in the scene (e.g. when you had to cut the rope or approach the genie)?

2. In your opinion, which test shooting mode is preferable for working on gaze direction? (e.g. when you had to look at the dog, the mini-genius or the presents)
3. In your opinion, which shooting test mode is preferable to work on synchronisation with the other characters? (e.g. when you had to react to events such as the dog leaving)
4. In your opinion, which shooting rehearsal mode is preferable for working on the emotional involvement in the scene (e.g. for emotional reactions required by the character being played, such as fright at the dog or astonishment at the genie's transformation)
5. Which shooting test mode do you prefer?

6.6.6 Augmented reality experience

These questions were taken from [88] and asked to rate the following statements on a scale from 1 (completely disagree) to 7 (completely agree).

1. Looking at virtual objects was as natural as looking at real-world objects.
2. I had the impression that virtual and real objects belonged to the same world.
3. I had the impression that, if I wanted to, I could touch and grasp the virtual objects.
4. I had the impression that the virtual objects were in the real world rather than simply projected on a screen.
5. I had the impression of seeing virtual objects as three-dimensional and not as mere flat images.
6. I noticed to the differences between real and virtual objects.
7. I had to make an effort to recognise virtual objects as three-dimensional.

Chapter 7

Results

7.1 Introduction

The experiment proposed was done to evaluate if and how augmented reality could be useful for actors to rehearse scenes in which the motion capture is involved. Two types of data were collected for each volunteer: subjective and objective. To study the statistical significance, a parametric student's t-test with a value of significance $\alpha < 0.05$ was performed on the results. In the next two sections, the data collected in the experiment will be analyzed in order to find out if augmented reality can be a valid alternative for actor's rehearsal.

7.2 Participants

Fifteen participants (aged between 21 and 45, mean=26,375 std=5,75) of which nine male and six female took part in the experiment. Seven of them were trained using Augmented reality while the remaining were trained using the traditional method. Regarding their level of knowledge of acting, most of them had minimal or low knowledge, 20% had some knowledge and 7% had good knowledge (Fig. 7.1). None of them had ever acted in scenes requiring the use of motion capture.

Participants were volunteers who received no compensation and were recruited through professional and personal contacts.

7.3 Subjective Results

The subjective measurements were collected through the two questionnaires presented above. As said in the previous chapter, the first questionnaire, included two items about spatial presence and social presence, one item about the usability of

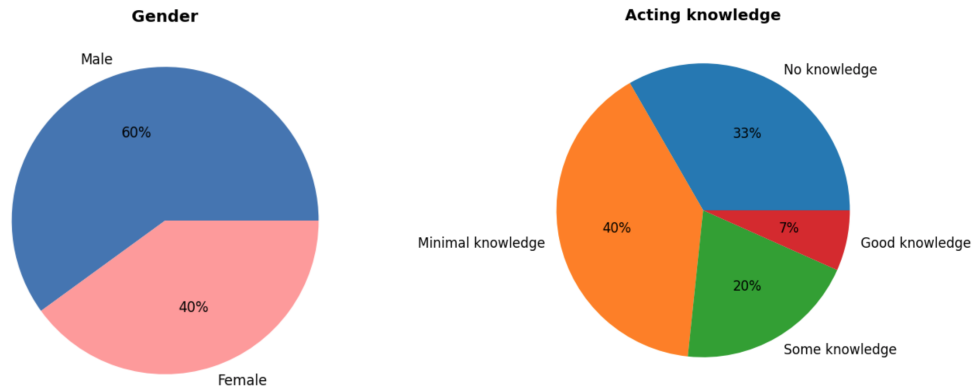


Figure 7.1: Features of the volunteers who participated in the experiment.

the rehearsal system, and a series of questions on how the rehearsal system helped the tester when shooting the scene. The second questionnaire instead concerned the rehearsal method preferences. Participants also had to answer a series of questions concerning their experience with augmented reality. For system usability, the SUS scores of the two rehearsal methods were compared (Fig. 7.2).

The AR method scored 90.4 while the traditional method 59.1. The participants found the AR method for rehearsal easier to learn than the traditional one, and they felt more confident in doing the rehearsal using this method. The other three parts of the questionnaire (concerning spatial presence, social presence and how the rehearsal method helped) collected subjective data with a 1 to 7 Likert scale. The results are illustrated in Fig. 7.3.

In all the three categories augmented reality rehearsal seems to be better for the participants. More specifically, regarding spatial presence, most of the participants stated that they felt most strongly that they were within the environment of the scene and that they felt more like they could reach out and touch the objects in the scene. Speaking about the social presence the participants declared that, using the AR method, they had more of a feeling of being able to interact with virtual characters (both animated in motion capture and not) and that they had more of a feeling that those characters could also see/hear them. Finally, the participants stated that the AR rehearsal method were more helpful for them in order to shoot the scene after the three rehearsal having no clue of the virtual object in the scene. The answers to the questionnaire showed that the AR rehearsal method helped them to make better use of space, to position themselves better in the scene, to feel more comfortable with their gestures, to manifest emotional reactions as a result of events involving virtual objects in the scene and, finally, to feel more confident when shooting the scene. All of these results were statistically validated through a student's t-test with a value of significance of $\alpha < 0.05$. In the

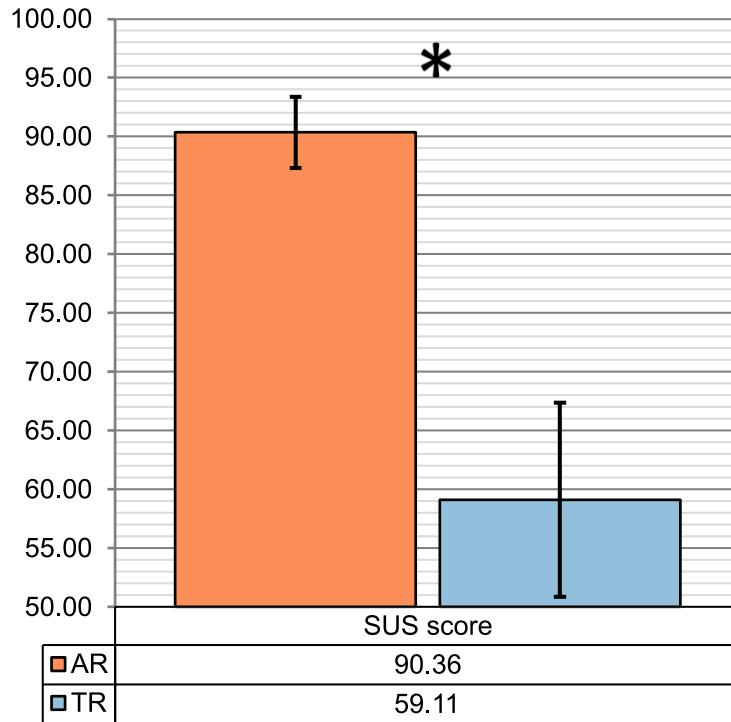


Figure 7.2: SUS score results (the higher the better).

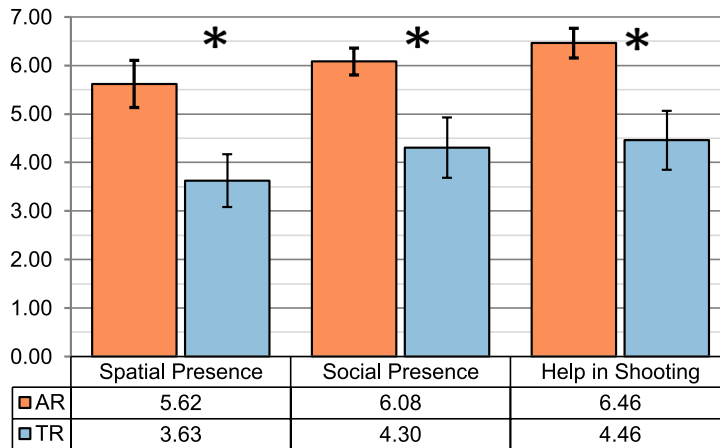


Figure 7.3: Questionnaire results (the higher the better).

second part of the questionnaire, concerning preferences, all participants stated that they preferred the augmented reality testing method. The p-values for the subjective data are reported in Table 7.1.

	P-Value
Usability of the system	<0.001
Spatial presence	0.003
Social presence	0.004
Help in shooting	0.002

Table 7.1: P-values regarding the subjective data.

7.4 Objective Results

The objective results of the experiment concerned two aspects: the eye-gaze and the spatial position. The data was collected during the shoot, when the volunteers had no visual clues about the virtual objects (including the Genie animated in motion capture), and was about the distance from where the testers were looking and where they had to look, and the distance from where they had to position their hand and where they actually position them. This data was collected using the HoloLens position/rotation for the tester's head position/rotation and a glove with Optitrack trackers for the tester's hand tracking. This data was analyzed considering specific moments during which the user had to look at a specific virtual object/character or had to place his or her hand on a specific virtual object. For eye-gaze, the value for the distance was taken as that between the point to be observed and the point of intersection of the user's line of gaze direction and the plane perpendicular to the segment joining the user's position with that of the point to be observed (as depicted in Fig. 7.4).

The collected data concerned:

- the eye-gaze distance calculated when the tester had to look at the dog for the first time at the beginning of the script;
- the eye-gaze distance's mean calculated when the tester had to look at the dog terrified, as it was walking towards him or her
- the eye-gaze distance's mean calculated when the tester had to look at the Genie's eyes while he was transforming in mini-Genie;
- the eye-gaze distance's mean calculated when the tester had to interact with the mini-genie for 10 seconds after transformation;
- the eye-gaze distance's mean calculated when the tester had to look at the diamond when it appears animating in a spiral movement;
- the eye-gaze distance's mean calculated when the tester had to look at the gem when it appears animating in a spiral movement;

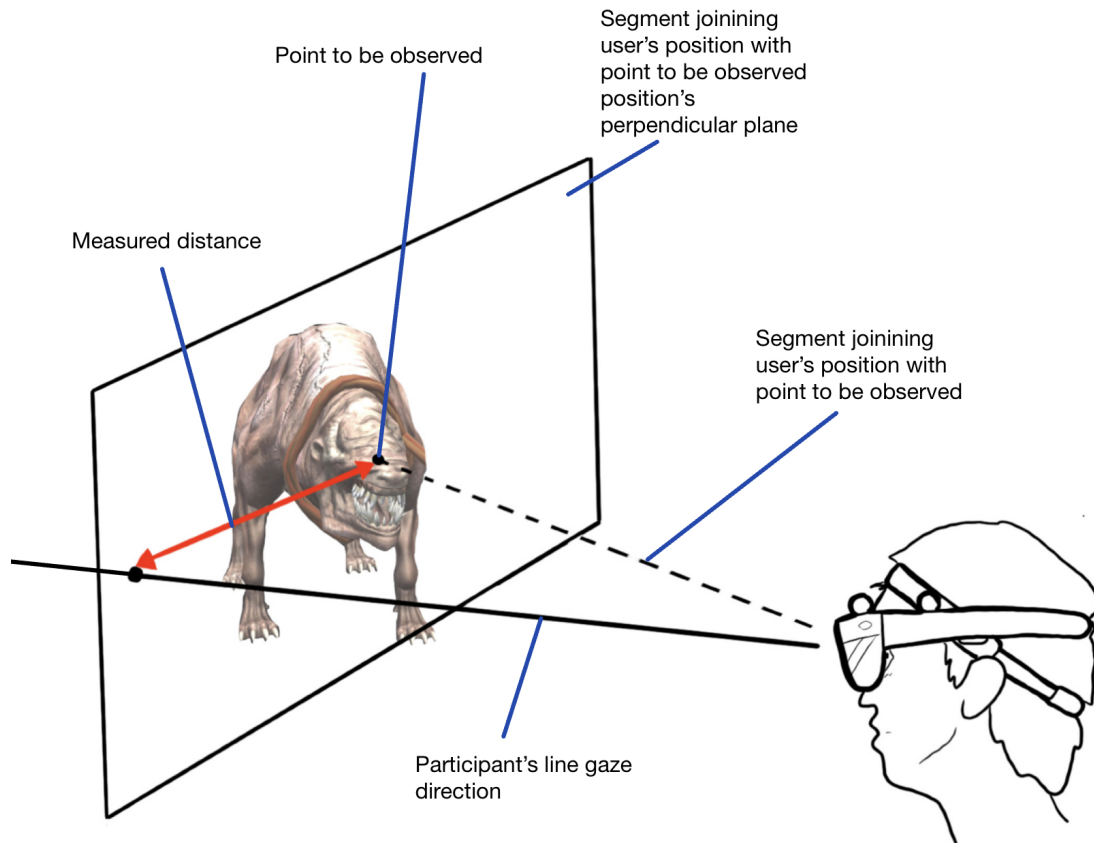


Figure 7.4: An illustration of how the distance was calculated for objective metrics.

- the hand distance calculated when the user had to take the gem at the end of the script.

Speaking about the eye-gaze distance data, as it can be seen in Fig. 7.5, all distances were shorter for the shoot scene done after the AR rehearse.

However, statistically, only three of these values were relevant: the one referred to the moment in which the participant had to look at the dog who was walking towards him or her, the one referred to the moment when the participant had to look at Genie's eyes during his transformation and the one referred to the moment in which the user had to interact with the mini-Genie after his transformation. For the first two, it is worth noticing that they represent two similar events: two animations of virtual characters that the participants had to follow with their eyes and react. The third is referred to an interaction with a virtual character animated through motion capture who's size is smaller than the human one. For the hand distance, the data collected showed that this was slightly lower when the scene

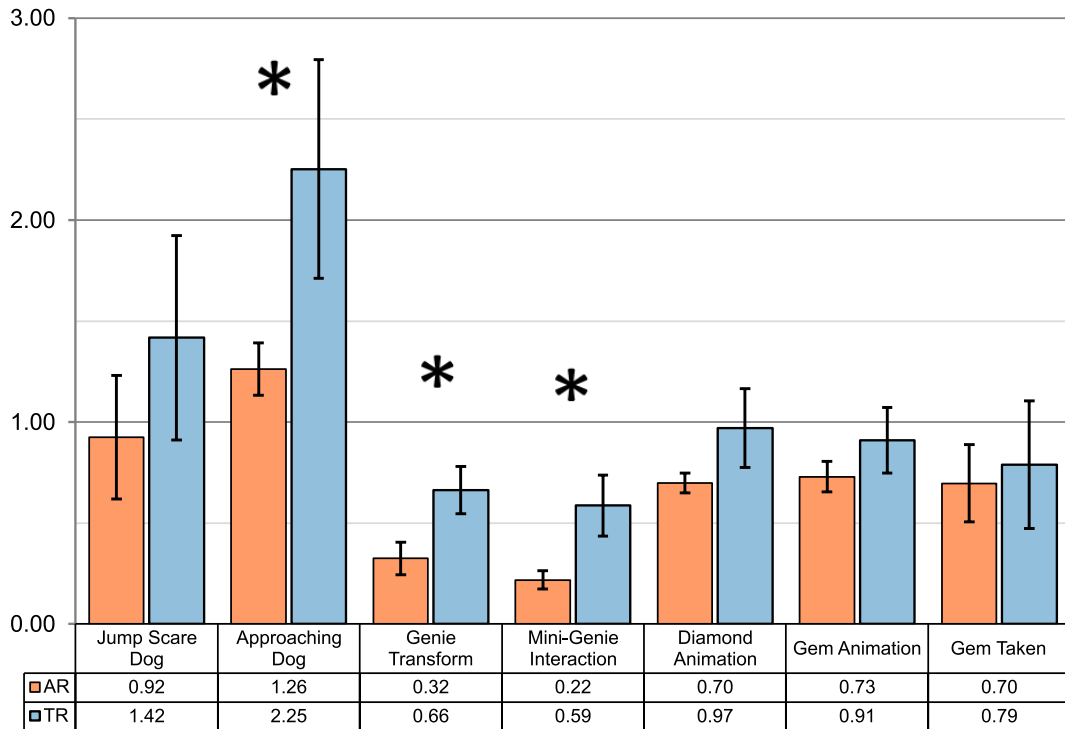


Figure 7.5: Data measured when shooting the scene after AR or traditional rehearsal (the lower the better).

was prepared with the AR rehearsal method than with the traditional rehearsal method (0.70 instead of 0.79). However, this result was not statistically significant. In Table 7.2 the p-values of the statistically significant elements are shown.

	P-Value
Approaching Dog	0.035
Genie Transform	0.007
Mini-Genie Interaction	0.008

Table 7.2: P-values regarding the subjective data.

So the objective data showed that the AR rehearsal can be a valid method for actors to train themselves in directing their gaze towards virtual elements in the scene as they move or to better direct their gaze when interacting with animated characters in motion capture of a different scale from the human one.

Chapter 8

Conclusions

The goal of this thesis is to analyze if and how AR could be helpful for actors to overcome their difficulties when playing scenes in which motion capture is involved. A framework was created to allow a user wearing the HMD to see the virtual character superimposed in real time on the actor who is animating it through motion capture.

This was done using an HoloLens 1st Gen. as OST-HMD, Optitrack as optical tracking system for motion capture and an open-source middleware called MotionHub to allow these systems to communicate with each other, by modifying the MotionHub software and creating a Unity application for HoloLens.

Afterwards, an experiment was carried out involving volunteers, regarding how useful AR can be in rehearsing scenes in which virtual elements, including characters animated through motion capture, are present. The aim of the experiment was to compare the proposed scene rehearsal method (with the use of the AR visor and the devised framework mentioned above) with the traditional rehearsal method involving the use of props.

Subjective and objective data were gathered through this experiment and analyzed. The obtained results showed that the AR method was better for the users in terms of usability, spatial and social presence in the scene, and the usefulness of the method for then filming the scene. With regard to the objective data, they showed how the AR trial can help in directing the gaze towards virtual elements in the scene as these elements move, or to better direct it when interacting with motion capture-animated characters of a different scale from the human one.

In conclusion, it was proved how AR can be a viable alternative for actors to rehearse scenes in which motion capture is involved.

The work presented in this thesis could be improved in many ways in the future. First of all, this framework can be extended to be used by multiple users, so that it could also be useful in scenes where there are several actors shooting a scene with one other actor playing a virtual character using motion capture. Secondly, if

a markerless system is adopted, the motion capture suit-wearing actor would be able to move in larger spaces and without the requirement for intrusive equipment like reflective marker outfits. Unmarked systems such as inertial ones could be used for this purpose, although latency problems would have to be reduced and the problems concerning differences between reference systems would have to be resolved somehow. Thinking bigger, the video camera integrated in the OST-HMD could be directly used to detect, by means of machine learning algorithms for example, the 3D pose of the actor playing the virtual character, and use it to overlay the character played.

Bibliography

- [1] Bridget Poetker. *A Brief History of Augmented Reality (+Future Trends & Impact)*. <https://www.g2.com/articles/history-of-augmented-reality>. 2019 (cit. on p. 2).
- [2] Andrew Makarov. *10 Augmented Reality Trends of 2022: A Vision of Immersion*. <https://mobidev.biz/blog/augmented-reality-trends-future-ar-technologies>. 2022 (cit. on p. 3).
- [3] Alessandro Evangelista, Lorenzo Ardito, Antonio Boccaccio, Michele Fiorentino, Antonio Messeni Petruzzelli, and Antonio E. Uva. «Unveiling the technological trends of augmented reality: A patent analysis». In: *Computers in Industry* 118 (2020), p. 103221. ISSN: 0166-3615. DOI: <https://doi.org/10.1016/j.compind.2020.103221>. URL: <https://www.sciencedirect.com/science/article/pii/S0166361519310899> (cit. on p. 3).
- [4] José Negrillo-Cárdenas, Juan-Roberto Jiménez-Pérez, and Francisco R. Feito. «The role of virtual and augmented reality in orthopedic trauma surgery: From diagnosis to rehabilitation». In: *Computer Methods and Programs in Biomedicine* 191 (2020), p. 105407. ISSN: 0169-2607. DOI: <https://doi.org/10.1016/j.cmpb.2020.105407>. URL: <https://www.sciencedirect.com/science/article/pii/S0169260719309794> (cit. on p. 3).
- [5] Rosanna M. Viglialoro, Nicola Esposito, Sara Condino, Fabrizio Cutolo, Simone Guadagni, Marco Gesi, Mauro Ferrari, and Vincenzo Ferrari. «Augmented Reality to Improve Surgical Simulation: Lessons Learned Towards the Design of a Hybrid Laparoscopic Simulator for Cholecystectomy». In: *IEEE Transactions on Biomedical Engineering* 66.7 (2019), pp. 2091–2104. DOI: 10.1109/TBME.2018.2883816 (cit. on p. 3).
- [6] «The effect of Augmented Reality Technology on middle school students' achievements and attitudes towards science education». In: *Computers & Education* 144 (2020), p. 103710. ISSN: 0360-1315. DOI: <https://doi.org/10.1016/j.compedu.2019.103710>. URL: <https://www.sciencedirect.com/science/article/pii/S0360131519302635> (cit. on p. 3).

-
- [7] Yong-Chin Tan, Sandeep R. Chandukala, and Srinivas K. Reddy. «Augmented Reality in Retail and Its Impact on Sales». In: *Journal of Marketing* 86.1 (2022), pp. 48–66 (cit. on p. 3).
- [8] F. Obermair, J. Althaler, U. Seiler, P. Zeilinger, A. Lechner, L. Pfaffeneder, M. Richter, and J. Wolfartsberger. «Maintenance with Augmented Reality Remote Support in Comparison to Paper-Based Instructions: Experiment and Analysis». In: *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*. 2020, pp. 942–947. DOI: 10.1109/ICIEA49774.2020.9102078 (cit. on p. 3).
- [9] <https://www.windowblogitalia.com/2021/05/hololens-microsoft-hanami/>. [accessed: 10-06-2022] (cit. on p. 4).
- [10] Kore. *The human eye’s understanding of space for Augmented Reality*. <https://uxdesign.cc/human-eyes-understanding-of-space-for-augmented-reality-d5ce4d9fa37b>. 2018 (cit. on p. 4).
- [11] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. «Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age». In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332. DOI: 10.1109/TR0.2016.2624754 (cit. on p. 5).
- [12] Andreas Jakl. *Basics of AR: Anchors, Keypoints & Feature Detection*. <https://www.andreasjakl.com/basics-of-ar-anchors-keypoints-feature-detection/>. 2018 (cit. on p. 5).
- [13] Sarah Rowe. *Spatial Mapping Explained: Your Guide to How Augmented Reality Works*. <https://circuitstream.com/blog/spatial-mapping/>. 2018 (cit. on p. 6).
- [14] Q. Wang D. Cheng Y. Wang H. Hua and G. Jin. *Design, tolerance, and fabrication of an optical see-through head-mounted display with free-form surface elements*. <https://opg.optica.org/ao/fulltext.cfm?uri=ao-52-7-C88&id=249592//>. 2013 (cit. on p. 7).
- [15] Yuta Itoh, Tobias Langlotz, Jonathan Sutton, and Alexander Plopski. «Towards Indistinguishable Augmented Reality: A Survey on Optical See-through Head-Mounted Displays». In: *ACM Comput. Surv.* 54.6 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3453157. URL: <https://doi.org/10.1145/3453157> (cit. on pp. 8, 9).
- [16] Mike Wiemer and Renaldi Winoto. «Mojo Lens - AR Contact Lenses for Real People». In: *2021 IEEE Hot Chips 33 Symposium (HCS)*. 2021, pp. 1–56. DOI: 10.1109/HCS52781.2021.9567321 (cit. on p. 10).

- [17] <https://www.mojo.vision/mojo-lens>. [accessed: 08-06-2022] (cit. on p. 10).
- [18] Laura Karreman. «The Motion Capture Imaginary: Digital renderings of dance knowledge.» In: (2017). URL: https://www.researchgate.net/publication/316428528_The_Motion_Capture_Imaginary_Digital_renderings_of_dance_knowledge/citations (cit. on p. 11).
- [19] Sharma, Shubham & Verma, and Shubhankar & Kumarand Mohit & Sharma Lavanya. «Use of Motion Capture in 3D Animation: Motion Capture Systems, Challenges, and Recent Trends.» In: (2019), pp. 289–294. DOI: 10.1109/COMITCon.2019.8862448 (cit. on p. 12).
- [20] Colombo, Giorgio & Facchetti, Giancarlo & Rizzi, and Caterina. «Virtual Testing Laboratory for Lower Limb Prosthesis. Computer-Aided Design and Applications.» In: (2013), pp. 671–683 (cit. on p. 12).
- [21] <https://optitrack.com/>. [accessed: 11-04-2022] (cit. on p. 13).
- [22] <https://mountcg.com/how-do-mocap-suits-work/>. [accessed: 01-02-2022] (cit. on p. 14).
- [23] <https://kknews.cc/news/jlpbm3y.html>. [accessed: 08-05-2022] (cit. on pp. 14, 15).
- [24] <https://www.cinefilos.it/cinema-news/news2018b/marvel-studios-backstage-foto-371950>. [accessed: 17-04-2022] (cit. on p. 15).
- [25] <https://aestheticcomplexity.wordpress.com/2020/04/20/mocap-fractals/>. [accessed: 14-06-2022] (cit. on p. 15).
- [26] <https://lujuba.cc/en/386233.html>. [accessed: 01-02-2022] (cit. on p. 16).
- [27] <https://www.youtube.com/watch?v=abxZPFThC4Q>. [accessed: 10-06-2022] (cit. on p. 16).
- [28] <https://epicstream.com/article/avengers-endgame-an-inside-look-at-how-they-turned-mark-ruffalo-into-smart-hulk>. [accessed: 09-05-2022] (cit. on p. 16).
- [29] <https://www.youtube.com/watch?v=LH18nGoIUKo>. [accessed: 10-02-2022] (cit. on p. 17).
- [30] <https://www.youtube.com/watch?v=UVLCZopu00A&t=26s>. [accessed: 05-02-2022] (cit. on p. 17).
- [31] <https://www.youtube.com/watch?v=a--pm7jtSck>. [accessed: 14-04-2022] (cit. on p. 17).
- [32] <https://www.xsens.com/motion-capture>. [accessed: 10-04-2022] (cit. on p. 18).

- [33] Bellusci G. Schepers HM Giuberti M. «Xsens MVN: Consistent Tracking of Human Motion Using Inertial Sensing». In: (2018) (cit. on pp. 18, 19).
- [34] <https://www.xsens.com/news/xsens-motion-capture-for-the-one-and-only-ivan> (cit. on pp. 19, 33).
- [35] <https://www.xsens.com/cases/ted>. [accessed: 13-04-2022] (cit. on p. 19).
- [36] https://www.youtube.com/watch?v=Zp26wNPI_hU&t=34s. [accessed: 15-03-2022] (cit. on p. 19).
- [37] <youtube.com/watch?v=8EnHFTb3yDc>. [accessed: 10-02-2022] (cit. on p. 19).
- [38] <http://www.andrew-whitehurst.net/pipeline.html>.. [accessed: 21-03-2022] (cit. on p. 21).
- [39] Noah Kadner. «The Virtual Production Field Guide, Volume 1». In: (2019) (cit. on pp. 22, 23, 25, 27, 34).
- [40] Noah Kadner. «The Virtual Production Field Guide, Volume 1». In: (2021) (cit. on pp. 23, 31).
- [41] <https://www.youtube.com/watch?v=rWtBH5LXEec>. [accessed: 21-03-2022] (cit. on p. 24).
- [42] https://techcrunch.com/2020/02/20/how-the-mandalorian-and-ilm-invisibly-reinvented-film-and-tv-production/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAANnKeF6XjPH_nua2GMORxpM3rfW-pzidVy5yITyjjlRHRru4nwKro10_3KS98FK15fF9Gp4x_m8-tAn4laSGzWzXfuhuU0zrl1dh7mjastw_KuaJqLo3jVFjMR1Q1S5VTKhD0mCaf_14i0QfMQkWVE_oSFBSYc0uEprK0ih4vwjX (cit. on p. 25).
- [43] <https://postperspective.com/tag/sony-crystal-led-displays/>. [accessed: 01-03-2022] (cit. on p. 25).
- [44] Marcus Dempwolf Holm. «The structuring of a VFX Pipeline». In: (2018) (cit. on p. 26).
- [45] Taylor Hoboken and Francis. «Production Pipeline Fundamentals for Film and Games». In: (2014) (cit. on pp. 26, 28, 29).
- [46] <https://www.youtube.com/watch?v=gvqGaDqwp88>. [accessed: 11-04-2022] (cit. on p. 28).
- [47] R. Ge and T. -C. Hsiao. «A Summary of Virtual Reality, Augmented Reality and Mixed Reality Technologies in Film and Television Creative Industries». In: (2020), pp. 108–111. DOI: 10.1109/ECBIOS50299.2020.9203607. (cit. on p. 30).
- [48] <youtube.com/watch?v=1uvSTbkBZi0>. [accessed: 10-05-2022] (cit. on p. 30).

- [49] <https://youtu.be/er4SgnjDYMI>. [accessed: 18-03-2022] (cit. on p. 30).
- [50] [youtube.com/watch?v=4NU9ikjqjC0](https://www.youtube.com/watch?v=4NU9ikjqjC0). [accessed: 18-03-2022] (cit. on p. 31).
- [51] https://www.youtube.com/watch?v=smj8i1__bmo. [accessed: 18-03-2022] (cit. on p. 31).
- [52] <https://www.youtube.com/watch?v=4NU9ikjqjC0>. [accessed: 19-03-2022] (cit. on p. 31).
- [53] [youtube.com/watch?v=W_6vTqIyPmM](https://www.youtube.com/watch?v=W_6vTqIyPmM). [accessed: 19-03-2022] (cit. on p. 31).
- [54] <https://www.youtube.com/watch?v=6z7wRntU464>. [accessed: 19-03-2022] (cit. on p. 32).
- [55] www.youtube.com/watch?v=a--pm7jtSCK. [accessed: 20-03-2022] (cit. on p. 32).
- [56] <https://www.insider.com/avengers-infinity-war-thanos-without-special-effects-photos-2018-7>. [accessed: 21-03-2022] (cit. on p. 33).
- [57] <https://www.thrillist.com/entertainment/nation/beauty-and-the-best-movie-dan-stevens-special-effects>. [accessed: 22-03-2022] (cit. on p. 33).
- [58] Xiaoming Chen, Zhibo Chen, Ye Li, Tianyu He, Junhui Hou, Sen Liu, and Ying He. «ImmerTai: Immersive Motion Learning in VR Environments, Journal of Visual Communication, and Image Representation». In: 58 (2019), pp. 416–427. ISSN: 1047-3203. URL: <https://doi.org/10.1016/j.jvcir.2018.11.039> (cit. on pp. 35–37, 45).
- [59] Juan Liu, Yawen Zheng, Ke Wang, Yulong Bian, Wei Gai, and Dingyuan Gao. «A Real-time Interactive Tai Chi Learning System Based on VR and Motion Capture Technology». In: 174 (2020), pp. 712–719. ISSN: 1877-0509. URL: <https://doi.org/10.1016/j.procs.2020.06.147> (cit. on pp. 35–37, 45).
- [60] Thuong N. Hoang, Martin Reinoso, Frank Vetere, and Egemen Tanin. «Onebody: Remote Posture Guidance System using First Person View in Virtual Environment». In: (2016). URL: <https://doi.org/10.1145/2971485.2971521> (cit. on pp. 37, 45).
- [61] Ikeda A., D. Hwang, and H. & Koike. «AR based Self-sports Learning System using Decayed Dynamic TimeWarping Algorithm». In: (2018) (cit. on pp. 37, 38, 45).
- [62] Damian, I., M. Obaid, F. Kistler, and E. & André. «Augmented Reality Using a 3D Motion Capturing Suit». In: (2013), pp. 233–234 (cit. on pp. 37, 45).
- [63] Stamm, A., and G. Teall P. & Benedicto. «Augmented virtuality in real time for pre-visualization in film». In: (2016) (cit. on pp. 38, 39, 45).

- [64] S. Ikeda et al. «CReal-time outdoor pre-visualization method for videographers—real-time geometric registration using point-based model—». In: (2008), pp. 949–952. DOI: 10.1109/ICME.2008.4607593 (cit. on pp. 39, 45).
- [65] T. Tamura. «Computer Vision Technology Applied to MR-Based Pre-visualization in Filmmaking. In Computer Vision». In: (2011), pp. 1–10 (cit. on pp. 39, 40, 45).
- [66] Ryosuke Ichikari, Ryuhei Tenmoku, F. Shibata, and Toshikazu Ohshima & H. Tamura. «Mixed Reality Pre-visualization for Filmmaking: On-set Camera-work Authoring and Action Rehearsal». In: (2008) (cit. on pp. 39, 40, 45).
- [67] Berthelot, Rozenn & Arnaldi, Bruno & Gouranton, and Valérie. «Virtual Reality Rehearsals for Acting with Visual Effects». In: (2016) (cit. on pp. 41, 42, 45, 81, 84, 86, 97, 100).
- [68] Robin K. Kammerlander, André Pereira, and Simon Alexanderson. «Using Virtual Reality to Support Acting in Motion Capture with Differently Scaled Characters». In: (2021), pp. 402–410. DOI: 10.1109/VR50410.2021.00063 (cit. on pp. 42, 44, 45, 78, 79, 84).
- [69] P. Ladwig, K. Evers, E. Jansen, B. and Nowotnik Fischer, D., and C. & Geiger. «Middleware for Unification of Multiple Body Tracking Systems». In: (2020) (cit. on pp. 47, 58).
- [70] <https://optitrack.com/applications/movement-sciences/>. [accessed: 18-04-2022] (cit. on p. 52).
- [71] <https://www.youtube.com/watch?v=miK3YqNsskE>. [accessed: 18-04-2022] (cit. on p. 52).
- [72] <https://optitrack.com/cameras/primex-13w/>. [accessed: 18-04-2022] (cit. on p. 52).
- [73] https://v22.wiki.optitrack.com/index.php?title=Cabling_and_Wiring. [accessed: 18-04-2022] (cit. on p. 53).
- [74] <https://uclalemur.com/blog/optitrack-motion-capture-system-with-crazyflie>. [accessed: 18-04-2022] (cit. on p. 54).
- [75] Richard Roesler. «A Guide to Optical Motion Capture». In: (). URL: http://physbam.stanford.edu/cs448x/old/Optical_Motion_Capture_Guide.html (cit. on p. 55).
- [76] https://v22.wiki.optitrack.com/index.php?title=File:LatencyDiagram_20.png. [accessed: 18-05-2022] (cit. on p. 57).
- [77] Matthew Wright. «OpenSoundControl:AnEnablingTechnologyforMusicalNetworking». In: (2005). URL: <https://doi.org/10.1017/%20S1355771805000932> (cit. on p. 58).

- [78] <https://msd-makerspaces.gitbook.io/next-lab/augmented-reality/resources/platforms-hardware/microsoft-hololens>. [accessed: 23-05-2022] (cit. on p. 61).
- [79] https://images.anandtech.com/doci/10115/Sensor_bar.jpg. [accessed: 23-05-2022] (cit. on p. 62).
- [80] <https://cdn.mos.cms.futurecdn.net/mTbaRHiGkwhRy5FiaY4gk4.jpg>. [accessed: 23-05-2022] (cit. on p. 63).
- [81] https://v22.wiki.optitrack.com/index.php?title=NatNet:_Data_Types#:~:text=NatNet%20data%20is%20packaged%20mainly,accessing%20the%20actual%20tracking%20data.. [accessed: 04-05-2022] (cit. on p. 66).
- [82] Mar Gonzalez-Franco and Tabitha C Peck. «Avatar embodiment. towards a standardized questionnaire». In: *Frontiers in Robotics and AI* 5 (2018), p. 74 (cit. on p. 80).
- [83] Chad Harms and Frank Biocca. «Internal consistency and reliability of the networked minds measure of social presence». In: *Seventh annual international workshop: Presence*. Vol. 2004. Universidad Politecnica de Valencia Valencia, Spain. 2004 (cit. on p. 80).
- [84] Matthew Lombard, Theresa B Ditton, and Lisa Weinstein. «Measuring presence: the temple presence inventory». In: *Proceedings of the 12th annual international workshop on presence*. 2009, pp. 1–15 (cit. on pp. 80, 99).
- [85] Yiannis Georgiou and Eleni A Kyza. «The development and validation of the ARI questionnaire: An instrument for measuring immersion in location-based augmented reality settings». In: *International Journal of Human-Computer Studies* 98 (2017), pp. 24–37 (cit. on p. 80).
- [86] <https://www.youtube.com/watch?v=1at7kKzBYxI>. [accessed: 10-06-2022] (cit. on p. 89).
- [87] John Brooke. «SUS: A quick and dirty usability scale». In: *Usability Eval. Ind.* 189 (Nov. 1995) (cit. on p. 98).
- [88] Holger Regenbrecht and Thomas Schubert. «Measuring Presence in Augmented Reality Environments: Design and a First Test of a Questionnaire». In: *CoRR* abs/2103.02831 (2021). arXiv: 2103.02831. URL: <https://arxiv.org/abs/2103.02831> (cit. on p. 101).

Ringraziamenti

Vorrei spendere qualche riga per ringraziare chi, in questi anni è stato presente nella mia vita e ha contribuito positivamente alla mia crescita personale e professionale

In primo luogo vorrei ringraziare il Prof. Lamberti F. per la sua disponibilità e il suo impegno nell'aiutarmi in questo lavoro di tesi. Vorrei inoltre ringraziare i miei due supervisor: Cannavò A. e Praticò G. che mi hanno aiutato in questo percorso e mi hanno guidato sempre in maniera professionale e attenta nello svolgimento del progetto di tesi. Ho imparato molto da loro.

Grazie ai miei genitori, che mi hanno sempre supportato e mai fatto mancare nulla, in questo percorso universitario così come sempre nella vita, e ai miei fratelli Edo e Gianpi, per essere una fonte di ispirazione e due pilastri su cui poter sempre contare.

Infine vorrei ringraziare i miei amici, sia quelli incontrati durante questi anni a Torino, che si sono rivelati più importanti di quanto avrei mai potuto immaginare, sia quelli di sempre, anche se lontani, per essere la mia seconda famiglia.

Grazie di cuore a tutti voi.

