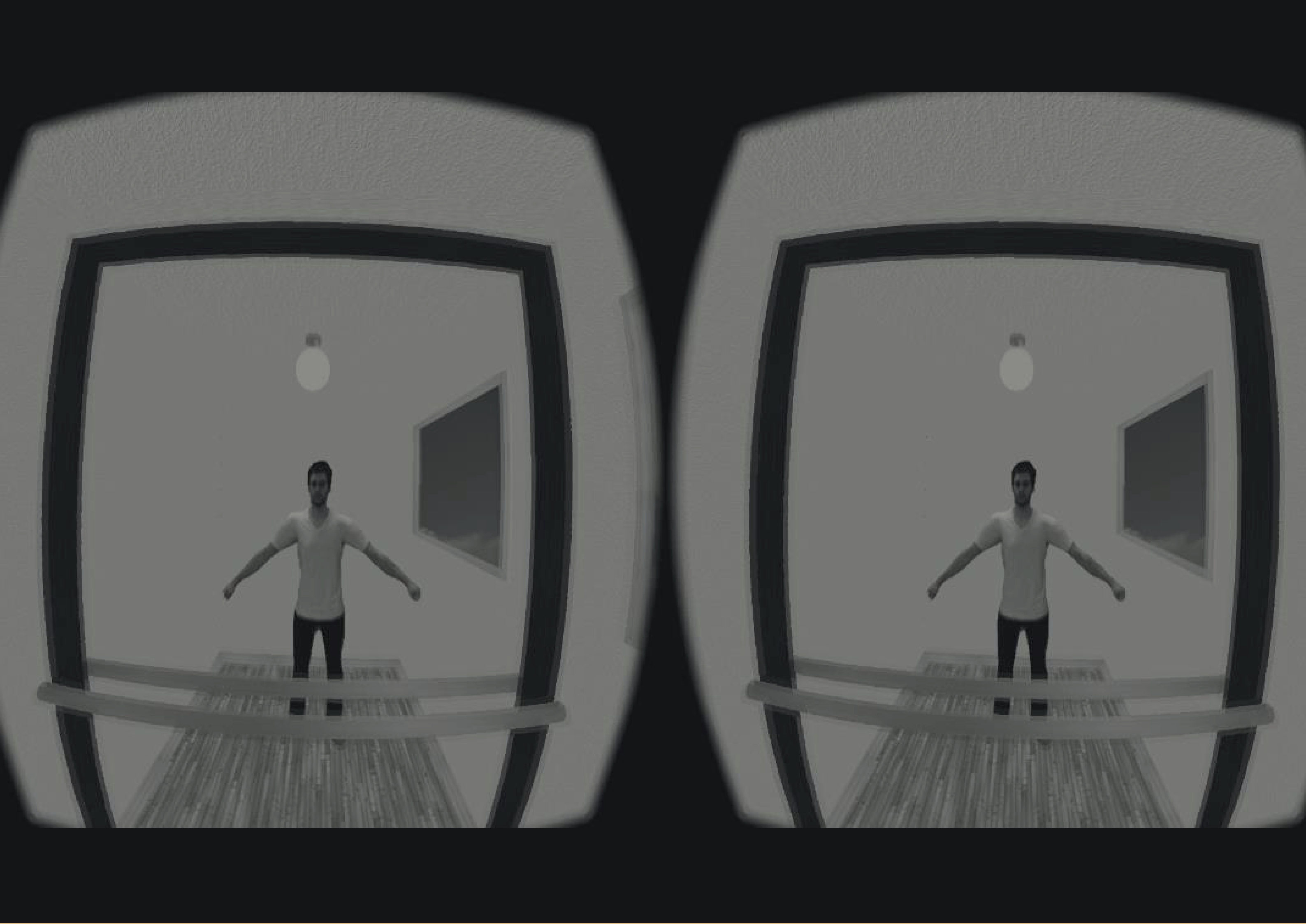

João Filipe Nunes Seabra

TRANSPORTING PEOPLE TO MIXED REALITY ENVIRONMENTS: EMBODIMENT EXPERIMENTS

Dissertação de Mestrado

Julho, 2015

· U C ·

Universidade de Coimbra

· U C ·

UNIVERSIDADE DE COIMBRA

Transporting people to mixed reality environments: Embodiment experiments

by

João Filipe Nunes Seabra

Orientador: Paulo Jorge Carvalho Menezes

Júri: Rui Paulo Pinto da Rocha

Nuno Miguel Mendonça da Silva Gonçalves

in the

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Electrotécnica e de Computadores

July 2015

"If you can dream it, VR can make it."

Matthew Schnipper

Abstract

The ability to safely travel to different worlds has long been a dream of mankind. Usually the user sits on a chair and experiences those virtual worlds through a screen that never amounts to more than a window to another universe. The evolution of head mounted displays (HMD) enabled the creation of systems capable of better transporting the user to these universes and many noticed their usefulness for scientific purposes. Teleoperation, psychotherapy, education and training simulators are some of the areas that profited from these mixed reality systems.

For these systems to be truly useful they must provide immersion and invoke a sense of embodiment in the user. Meaning that they must shut the user down from the real environment, provide an ample and clear view of the world, create a sense of ownership of the user's virtual representation and accurately transport the users, their likeness and their motions to that same virtual world.

This work will cover the development of a system designed to provide the best sense of embodiment and immersion while also being affordable and easy to setup, making it viable for being used in a clinical environment. For this effect, three main ingredients were necessary. An HMD and a body tracker were integrated and a working pipeline for scanning the user and creating his/her 3D model was found. The model is then animated in real time with data from the body tracker and the HMD's head tracker. The camera is placed on the avatar's head providing the user with a first person view of the world and allowing the user to look at his/her virtual avatar as he/she would in the real world.

With this system several demonstrations were developed. The first is a mirror room where the user can see the avatar and its reflection and served as a base to others. The second aimed at creating a traumatic experience while measuring physiological data. In that data it can clearly be seen that the users' biological signals spiked following this event indicating the systems success in its immersiveness. The third aimed at creating a tool to help the treatment of dysmorphic body image allowing the user to interact with a set of virtual buttons that scale certain parts of the avatar. The forth covered the use of an immersive system for teleoperation which was reported by the users to be more intuitive and natural than what is currently used.

Keywords: immersive system, mixed reality, embodiment, teleoperation, psychotherapy, 3D modelling, 3D animation.

Resumo

Há imenso tempo que a possibilidade de visitar outros mundos tem sido um sonho da humanidade. Normalmente o utilizador experiencia esses mundos virtuais através de um monitor, não sendo mais que uma janela para esse universo. A evolução dos *head mounted displays* (HMD) proporcionou a criação de sistemas capazes de melhor transportar o utilizador e muitos foram os que repararam na sua utilidade para a investigação científica. Teleoperação, psicoterapia, educação e simuladores de treino são algumas das áreas que beneficiaram destes sistemas de realidade mista.

Para estes sistemas serem realmente úteis, têm que proporcionar ao utilizador uma sensação de imersão e corporização. Isto significa que têm que obstruir o mundo real, permitir uma visualização ampla e clara do mundo virtual, criar uma sensação de propriedade sobre a representação virtual do utilizador e transportar o utilizador, a sua aparência e movimentos para esse mesmo mundo.

Este trabalho detalha o desenvolvimento de um sistema desenhado para proporcionar o melhor sentido de corporização e imersão mantendo um baixo custo e facilidade de configuração, tornando-o viável para uso em ambiente clínico. Para tal, são necessários três ingredientes. Um HMD e um *body tracker* foram integrados e foi encontrada uma forma de criar o modelo 3D dos utilizadores. Esse modelo é animado em tempo real com uma fusão da informação recolhida pelo *body tracker* e pelo *tracker* do HMD. A câmara é posicionada na cabeça do avatar proporcionando uma vista para o mundo em primeira pessoa e permitindo ao utilizador olhar para a sua representação da mesma forma que faria no mundo real.

Com este sistema foram desenvolvidas várias demonstrações. A primeira é um espelho onde o utilizador vê o seu avatar e a reflexão e serviu de base para as outras. A segunda trata-se da exposição do utilizador a um evento traumático enquanto os seus sinais biológicos são medidos. Nesses sinais podem ser claramente vistos picos de actividade após o evento traumático sendo um indicador da imersividade do sistema. O terceiro visa criar uma ferramenta para o tratamento de dismorfofobia permitindo ao utilizador interagir com um conjunto de botões que modificam o tamanho de certas partes do avatar. O quarto cobre a adição de um sistema imersivo para a teleoperação e foi reportado pelos utilizadores como mais intuitivo e natural que as alternativas actualmente usadas.

Palavras-chave: sistema imersivo, realidade mista, corporização, teleoperação, psicoterapia, modelação 3D, animação 3D.

Contents

List of Figures

Abbreviations

AR	**A**ugmented **R**eality
AV	**A**ugmented **V**irtuality
FoV	**F**ield **o**f **V**iew
HMD	**H**ead **M**ounted **D**isplay
OpenAR	**O**pen Source **A**ugmented **R**eality
OpenCV	**O**pen Source **C**omputer **V**ision
OpenGL	**O**pen Source **G**raphics **L**ibrary
OpenNI	**O**pen Source **N**atural **I**nterface
OSG	**O**pen **S**cene **G**raph
SDK	**S**oftware **D**evelopment **K**it
SDL	**S**imple **D**irectMedia **L**ayer
SoE	**S**ense **o**f **E**mbodiment
PTSD	**P**ost **T**raumatic **S**tress **D**isorder
RE	**R**eal **E**nvironment
VE	**V**irtual **E**nvironment
VR	**V**irtual **R**eality
VRET	**V**irtual **R**eality **E**xposure **T**herapy

Chapter 1

Introduction

The ability to safely travel to different worlds has long been a dream of mankind. Flying, visiting other planets, experience a zombie apocalypse or any other crazy dream or nightmare has been commonly available through movies and videogames for a long time. The user sits on a chair and experiences those virtual worlds through a screen that never amounts to more than a window to another universe.

Fortunately we are now entering a new era marked by the appearance of virtual reality (VR) in a more broad and mainstream context. New devices are now beginning to enter the mass market making immersive systems affordable and practical.

1.1 A Brief History of Immersive systems

The idea of immersing users in virtual experiences is not a recent one. We have been trying to create these systems since the mid 50s when the visionary Morton Heilig detailed the creation of a multi-sensory theatre called *Sensorama* in his paper *'The Cinema of the Future'* [1]. This device was envisioned to be able of producing stereo 3D images, body tilting and stereo sound. Even though this was a breakthrough in immersive system technology it lacked funding and was only truly produced later in 1962 as a working prototype where the user could experience a

3D video of a bicycle ride through Brooklyn. It may have lacked in success, but he planted the seed that served as inspirations to several other devices.

In 1961 Philco Corporation® created the first head mounted display (HMD) called *HeadSight*, complete with a screen and a magnetic tracking system. Unfortunately this device was only ever just directly connected to a camera in a closed circuit not fulfilling the objective of really transporting the user to another time and/or place. But it wasn't long until someone was able to achieve this goal. In 1968, Ivan E. Sutherland created the first fully capable HMD [2]. He used a mechanical arm hanging from the ceiling that connected to the HMD and tracked the head's movements. It may have been a cumbersome device but it possessed all the features we can see in modern HMDs including a connection to a computer for computer generated graphics.

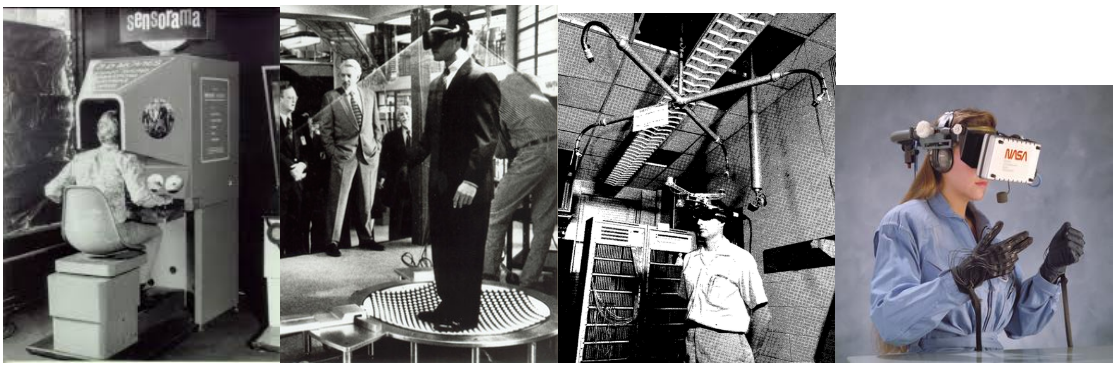

FIGURE 1.1: From left to right: Sensorama; HeadSight; Sutherland's HMD; NASA's immersive system

The year of 1985 marks the involvement of NASA in this field. They developed a system that combined an HMD and a pair of gloves capable of tracking the user's hands. This allowed for some degree of object manipulation and increased immersion. In the following decades the dream of VR went through some ups and downs. Several companies tried to bring these experience to the mass market. Nintendo® almost went bankrupt with the launch of Virtual Boy® in 1995 with the company completely dismissing its much marketed peripheral just an year after launch. In the same year Forte® launched its HMD called "VFX1" which became one of the most successful devices of this era but still not becoming the success many hoped. Marked by failures like this, VR was used for scientific research rather than being mass marketed. Despite this fact, the technology evolved steadily both in terms of field of view (FoV) and screen resolution.

FIGURE 1.2: From left to right: VirtualBoy; VFX1; Z800 3DVisor; Rift DK2

In late 2012, Palmer Luckey, the founder of Oculus VR®, started a Kickstarter campaign for The Rift. A modern day HMD that captured the attention of the world and especially the attention of Facebook® that proceeded to buy Oculus VR® for the historical sum of 2300$ million. VR is now a widely known concept and, with the recent involvement of every major company and the processing power of current smartphones and computers, is expected to be present in households all over the world in a short time.

The following table presents the evolution of HMD's in terms of resolution and field of view.

TABLE 1.1: Evolution of HMDs

HMD	Launch Year	Resolution per eye	diagonal FoV (degrees)
VFX1	1995	263*230	35.5
Virtual Boy	1995	1*224 scanned	unknown
Z800 3DVisor	2005	800*600	32.46
Rift DK1	2012	640*800	90
Rift DK2	2014	960*1080	90
OSVR	2015	960*1080	100

1.2 Applications of Immersive Systems

Immersive systems naturally started as recreational tools. Video games and 3D cinema were the first big motivations behind their development. As technology improved some researchers started to realize the potential of these systems for other more serious matters like psychology, teleoperation, education, training simulators, physical rehabilitation and occupational therapy (to name a few). The ability to progressively expose an user to a certain element without any real danger is indeed a huge advantage of these systems.

One of the most covered uses of VR is the treatment of post traumatic stress disorders (PTSD). Articles like [3] and [4] cover the use of immersive systems to treat PTSD caused by the Vietnam war and the World Trade Center event. In [5] Albert A. Rizzo details an analysis of the usage of VR systems for performing virtual reality exposure therapy (VRET). In all these publications the use of VRET has been reported to have great efficiency.

The work presented in [6] covers three case studies where VR was used to treat phantom limb pain reporting a significant decrease in pain felt during the experiences.

G. Riva, in [7], presents a comprehensive review of the usage of VR for psychotherapy covering, among others, treatments for fear of flying [8], acrophobia [9], arachnophobia [10], claustrophobia [11] and agoraphobia [12]. It also mentions the treatment of eating disorders and body image disturbances [13], [14]. This is where our work will be focused on. By following the teachings of researchers like Mel Slater for creating the desired sense of embodiment (SoE) we aimed to create a system ready to be used in a clinical environment with a focus on avatar deformation for the treatment of body image dysfunctions. With this in mind, the system was made to be easy to setup and affordable maintaining capability and effectiveness.

1.3 Contributions

The systems used for research, like the one used by Mel Slater [15], usually take advantage of high end HMD's and complex positional tracking hardware. In that specific publication he describes a system that uses 12 cameras and a special marker equipped suit to track body movements. These systems may be stabler and more precise but are far from viable in terms of practical use. They require a special room and too much setup. The contribution of this work is the creation of a system that is affordable and easy to setup, having in mind its usage in a clinical environment. However, several other applications were tested, mainly in the area of teleoperation.

This document will focus on the creation of several packages that enable all the needed functionality and the integration of all the required devices. Also, some demonstration experiences serving as tests for the system's immersiveness were developed.

Some of these demonstrations were just minor experiments that won't be mentioned and some were more fleshed out and even culminated in scientific publications.

The main experiences created are:

- The Mirror - consist on a virtual mirror where the user can see his/her actions recreated not just by his/her virtual body but also his/her reflection (fig. 4.1).
- The Guillotine - an experience designed to elicit fear and escape reactions while measuring physiological data to teach users about the effects that emotions play in our bodies (fig. 4.3).
- The Body Editor - similar to The Mirror but this experience allows the user to scale certain body parts and see those effects in real time on his/her virtual body (fig. 4.4).
- The Submarine - a project parallel to the main system, developed to facilitate the teleoperation of an underwater vehicle (fig. 4.7).

From these demos resulted the following scientific publications.

- B. Patrão, J. Seabra, S. Pedro, P. Menezes, An Affective Interaction System to Learn About Physiological Signals, Exp.at'2015 International Conference, 2015.
- B. Patrão, J. Seabra, S. Pedro, P. Menezes, Demonstration of the Influence of Human Emotions in Physiological Signals, Exp.at'2015 International Conference, 2015. (Honorable Mention Award)
- J. Garcia Sanchez, B. Patrão, J. Perez, J. Seabra, P. Menezes, J. Dias, P. Sanz, Towards an immersive and natural gesture controlled interface for intervention underwater robots, OCEANS'15 MTS/IEEE Genova, 2015.
- J. Seabra, B. Patrão, S. Pedro, P. Castilho and P. Menezes, Immersive Technologies as a Key Tool in Therapeutic Contexts, 4th International Conference On Compassion Focused Therapy, 2015 (submitted)
- J. Seabra, B. Patrão and P. Menezes, Be Yourself in Virtual Environments SciteclN'15 - Sciences and Technologies of Interaction, 2015 (submitted)

1.4 Organization

The contents of the following chapters is as follows. Chapter 2 (**From Virtual Reality to Immersion and Embodiment**) details the necessary ingredients to create immersion and Sense of Embodiment (SoE). Chapter 3 (**Developing an Immersive System**) covers the technicalities of the developed system. Chapter 4 (**Demonstrations and Case Studies**) presents a more detailed description of the most important demonstrations and experiences that were conducted, as well as some results and impressions. Chapter 5 (**Conclusion**) details the conclusions being taken from each experiment and also the general impression about this work. Some further improvements to the system are also mentioned.

1.5 Personal Motivation

I have forever been fascinated by the idea of creating meaningful interactive experiences and being able to work with devices as interesting as the Rift and Kinect to create these immersive experiences designed to help people in so many different ways is the motivation behind this work. Systems like the one that we set to build can facilitate many tasks. The only limit is the imagination of those that commit themselves to develop new and extraordinary experiences.

Chapter 2

From Virtual Reality to Immersion and Embodiment

Even though VR is the most commonly used term, Fumio Kishino [16] proposed a spectrum to better classify the type of an experience. This spectrum is presented in figure 2.1. In one end there is the real environment and, in the other end, the virtual environment, everything in between is classified as mixed reality. Inside this mixed space we have augmented reality (AR) closer to the real environment for being the juxtaposition of virtual elements on that same environment and a concept he called *'Augmented Virtuality'* (AV) which is the inclusion of real elements on what is mainly a virtual environment. The work developed along this thesis, according to F. Kishino is situated on the AV area of this spectrum since we will try to transport the user and his/her likeness to a virtual environment to the best extent possible. Nevertheless, the term Augmented Virtuality is infrequently used in the literature and replaced by mixed or augmented reality.

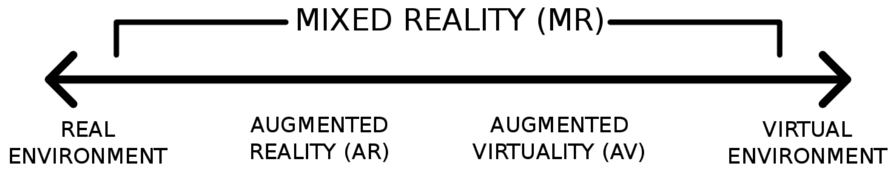

FIGURE 2.1: The spectrum of mixed realities, according with [16]

Despite the area of the spectrum where a mixed system may be situated, there is always the common objective of making the experience as credible and believable for the user as possible. It is important that the user perceives the virtual world (in the case of an AV system) or the virtual elements (in the case of an AR system) as natural parts of the experience. For this to happen the senses of immersion and presence must be invoked.

M. Slate et al., in [17], presents a distinction between immersion and presence. He describes immersion as being a quantifiable characteristic of the technology used that depends on four principles: inclusiveness, extensiveness, surroundings and vividness. Inclusiveness is the extent to which the user is shut down from the real environment, extensiveness is the range of body sensors being stimulated by the system, surroundings concern the wideness of the user's window to the virtual environment and vividness is about resolution and fidelity of the display used. This immersion, he states, is nothing without the ingredient he calls *'matching'*. This is the synchronization between the user's proprioceptive feedback and the sensory stimulation of the system. It is only natural that the immersion is lost if the user's vision to the virtual environment is not perfectly matched by his/her head movements, for example. Presence, on the other hand, is much more complicated to describe and implies the aforementioned immersion. It is a psychological state of consciousness of being in a space that implies that the user recognizes the virtual environment as a space he is visiting rather than an image on a screen. With this in mind it is only natural that this sensation is highly subjective. Different users can be more or less sensitive to the presented stimuli and have different responses.

To attain the full level of immersion, the user not only needs to perceive the surrounding space, but the own body in relation to that surrounding space. This importance of the perception of one's body is covered in [18]. This work presents a study showing that the representation of the user's body in an AV system influences the way that space is perceived. This representation of the user's body leads to the sense of presence and embodiment, meaning that we want the user to sense this virtual body as his/her own in order to accurately perceive the surrounding space.

To create this sense of presence and to make an user recognize the virtual environment as a space, it is mandatory that a sense of embodiment (SoE) is stimulated. In [19], Mel Slater et al. proposed a formal definition of what the SoE is. He describes this sense as a composition of three parts: sense of self-location, sense of agency and sense of body ownership. So, apart from any other stimuli, in order to be truly immersed (meaning, feeling present in the virtual environment) the user must be able to recognize a certain portion of the world's volume as his/her own, must feel like he/she has direct control of his/her virtual body and must recognize this virtual body as his/her own.

2.1 Towards embodiment sensations

Some systems, like virtual dressing rooms have also tried to explore this SoE. The user holds some kind of marker and the system overlays certain elements over the user's image trying to create a mirror-like experience. The problem with this system consists on the discontinuity between spaces. In that case, the virtual environment is perceived only though a screen. If the user looks down at himself/herself, the virtual clothes won't be there. We wanted to create a mixed reality system capable of providing better immersion and embodiment and for this, we tried to fully transport the user to the virtual environment.

For this effect, and following the work of Mel Slater, we needed three main ingredients: an HMD, a body tracker and a way to create the user's 3D model. The HMD serves as both a way of displaying the virtual environment to the user and shutting the user from the real environment, providing inclusiveness and extensiveness. It should also feature good surroundings and vividness as defined in [17]. To complete the immersion we wanted to provide further extensiveness. The user's sense of sight is already covered by the HMD but, since the virtual environment will be created on a 1:1 scale with the real one, the user will be able to touch his/her surroundings both physically and virtually providing tactile feedback. Some demonstrations will also make use of sound effects. All of this works towards the creation of the so desired immersion.

Now that the immersion is complete, a way of creating SoE must be found. Since for the successful creation of that SoE the user must feel in control of his/her virtual representation (or avatar), a body tracker must be used. With the inclusion of this sensor, the system is able to translate the user's motion in real time to the avatar and also position the avatar on the virtual environment according to the user's position on the real one.

As said above, the third element is the modelling of the user. With a 3D scanner we are able to completely transport the user's likeness to the virtual environment. This avatar is then animated with the data collected from the body tracker. And the camera to the virtual world is placed on the avatar's head.

Merging all this together we are able to put the user inside an avatar, gazing at the virtual environment through its eyes and moving its body just by moving his/her own, stimulating the sense of embodiment.

The proposed system architecture is shown below in figure 2.2. The user's model is created and passed to the scene generator. That model is then placed and animated with data from the body tracker. Information about the avatar's head position in the virtual environment is then fed to the view generator, that merges that data with the orientation from the head tracker included in the HMD. The result is our proposal at an immersive system for stimulating the SoE.

In figure 2.3 a man looking at a mirror is shown. The man can see both his arms and his reflection. This familiar experience is possible to virtually recreate with this system.

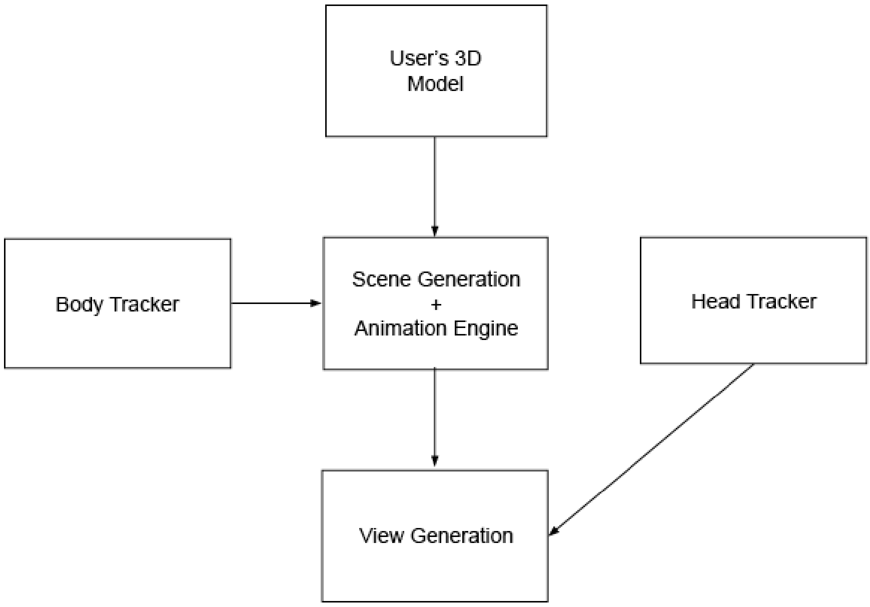

Figure 2.2: General system architecture

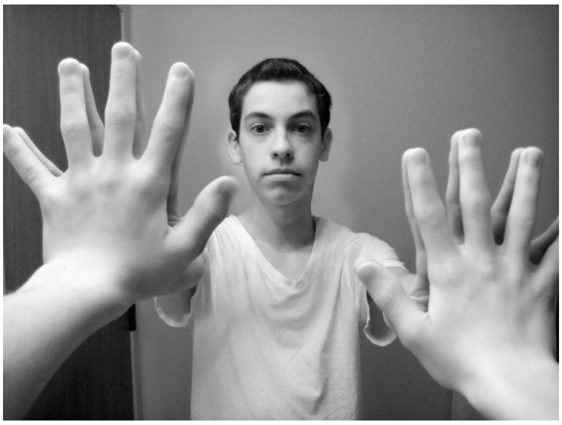

Figure 2.3: Person looking at mirror

2.2 Uses of embodiment

Recreational and serious games are one of the most obvious applications of these systems. They can serve several purposes ranging from education to therapeutic use. The fact that the user is led to feel a virtual avatar as his/her own body allows the experiences to invoke strong feelings. If the avatar is put in a dangerous situation, for example, and the user is effectively immersed in the experience it is natural to invoke anxiety and escape reactions. The work presented in [20] details how an user's brain reacts similarly to harmful events presented to a body which he/she has ownership illusion. This invoked anxiety is not only a consequence of the immersion, but also an enhancer of said immersion as detailed in [21].

This anxiety and the illusion of harm caused by the SoE can be fruitful in many applications. This work will focus on two main areas: teleoperation and psychotherapy. The benefit for teleoperation involves the improvement of the user's perception of the vehicle's surroundings making the control more natural and intuitive. For psychotherapy there is the possibility of creating interactive experiences in safe virtual environments.

Throughout this work several uses of the developed system and the embodiment it provides will be explored. Starting with a simple mirror experience for creating the SoE that aims at being as natural as possible. That experience and environment is then modified and applied to two other demonstrations. One is a harmful experience to test the user's reactions and the other aims at being a clinical tool for the treatment of biased body image and body dysmorphic disorder. The last experiment is a case study about the benefits of the application of these systems in the teleoperation of an underwater vehicle.

Chapter 3

Developing an Immersive System

Following the concepts presented in the previous chapter we propose an architecture that will support the development of the various types of applications.

From the very beginning, this work had the objective of creating an immersive system able to provide a good SoE without sacrificing availability and cost.

A Rift DK2 was made available for the purpose of this work, this HMD is one of the best mainstream HMDs currently available capable of producing 90° FoV and fullHD (1920*1080) total screen resolution. The human FoV is close to 180° but most of that Fov is peripheral vision and so the Rift covers the most important half of that FoV. Another important aspect is the fact that this HMD blocks the rest of the user's FoV, leaving the user focused on the image shown. These specifications are enough to create the desired inclusiveness since the user will be shut down from the real world unable to see anything apart from the Rift's screen, vividness because of the screen's quality and surroundings because of the big FoV being covered.

One of the most affordable and covered body trackers currently available is Microsoft's Kinect and as such, it is the one used in this work. It was launched on 2010 as a Xbox home entertainment system peripheral but saw great usage in scientific investigation for being a cheap and capable skeleton tracker.

For the 3D scanning of the user, Occipital's Structure sensor was used. This sensor connects with an iPad and, through the provided application, is able to produce highly detailed 3D models. The possibility of transporting the user to a virtual environment where the virtual body is his/her actual model greatly helps the SoE that is so desired. It is important to note that the models acquired through the Structure sensor are usually very high poly. Since these models are much more detailed then what is needed they are usually downsampled to have a lower polygon count. We tried this process both in Maya and Blender and found the later to produce better results. The downsample factor we found appropriate was about 1/3 of the original model. This factor still maintains the graphic detail needed for the user to easily recognize the avatar but also allows the experience to run smoother on less capable PCs and improve the 'matching' of the system. After the application is done collecting data and creating the model's mesh, it is only a matter of rigging the model (the rigging process is explained in section 3.4).

3.1 Setup and Framework

The development of this system required that everything would be built on top of a pre-existing in-house made framework called OpenAR (AR standing for augmented reality). This library contained several classes and functions usually found on video game engines. It was made to easily load models and textures to a virtual environment, create and manage windows, position virtual cameras, etc. To make all of this possible, OpenAR uses several known libraries, among those: OpenGL, SDL and OpenCV.

In order to use Oculus Rift and Kinect along this existing framework several libraries were included. OpenNI (open natural interaction) and NITE, both originally developed by PrimeSense, handle the motion tracking powered by Kinect. Oculus also made an SDK available with all the wrappers necessary to handle the specialized rendering needed for the Rift's usage along with the capturing of the head's yaw, pitch and roll. All of these libraries were included in OpenAR and several classes and wrappers were developed to acquire the needed data. An extensive overview of the developed modules is presented in the next subchapters and a schematic detailing the developed modules is presented in figure 3.1.

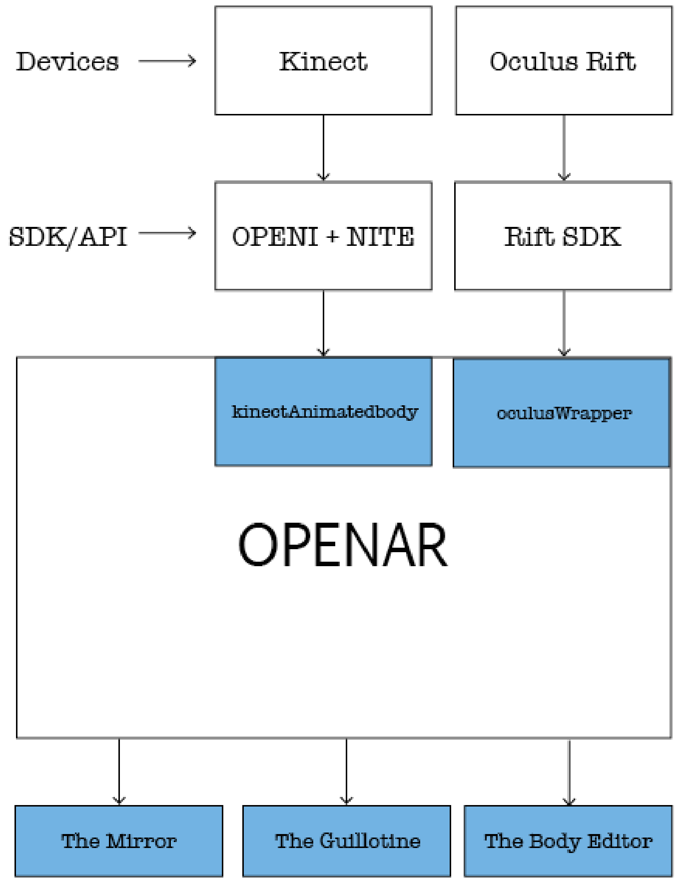

FIGURE 3.1: Presented in blue are the developed modules and how they relate to the rest of the framework

Our experience with the Kinect showed that it has some limitations tracking certain joints. Due to this fact, and to improve the fidelity of the captured head motion, the architecture presented in 2.2 was altered to include sensor fusion between the Rift's head tracker and neck rotation captured by the Kinect. This alteration is represented by the red arrow in the following figure 3.2.

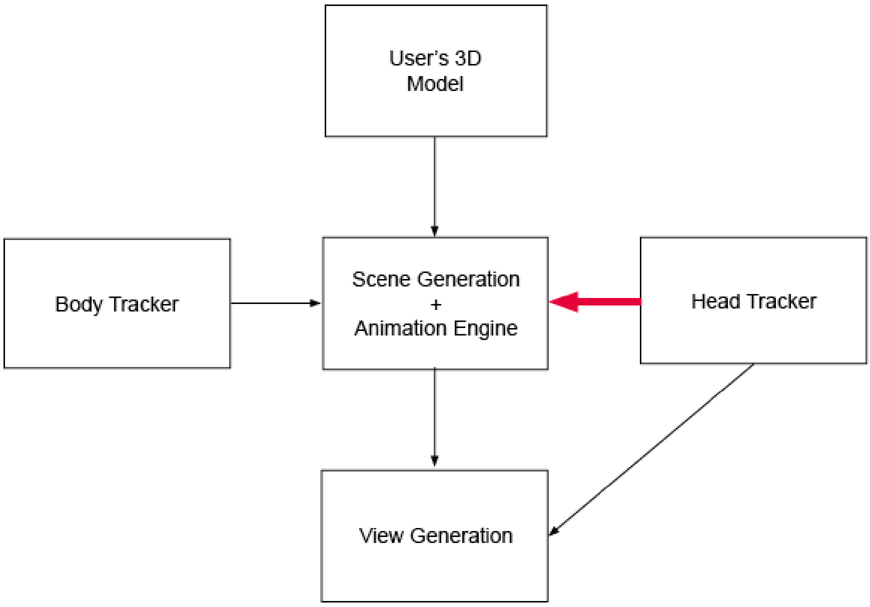

FIGURE 3.2: Architecture with sensor fusion

3.2 Oculus Rift

The Rift is a head mounted display (HMD) designed to create the most immersive experiences possible. It features a screen big enough to cover both the user's eyes and two lenses (one for each eye) made to expand the field of view (FOV). Each halve of the screen displays a slightly different image to create a believable 3D effect. Through a built-in gyroscope and an external camera, it is also capable of tracking the user's head position and orientation in 3D space. For each eye, a virtual camera is created, the textures rendered by those two cameras are then distorted to compensate the distortion of the lenses and are then displayed on the screen. The SDK made available by Oculus handles these intricate parts of the VR experience. The developer does not have to configure the interpupillary distance or the camera distortion because all of this is done by the SDK and the Rift's driver.

For the experiences developed along this work the head's position given by the Rift's head tracker will not be needed because this information will be gathered by the Kinect sensor and so, after correctly integrating the Rift's rendering in

OpenAR, the only thing left is to use the SDK to gather the head's yaw, pitch and roll angles and convert that to abide the OpenAR standard.

The first step is the conversion from the Euler angles provided by the Rift to a full rotation matrix. This conversion is achieved through the following equation with yaw $\rightarrow y$, pitch $\rightarrow p$, roll $\rightarrow r$, $c(\alpha)$ and $s(\alpha)$ referring to $cos(\alpha)$ and $sin(\alpha)$ respectively:

$$\mathbf{R}_{head} = \begin{bmatrix} c(y)c(p) & c(y)s(p)s(r) - s(y)c(r) & c(y)s(p)c(r) + s(y)s(r) \\ s(y)c(p) & s(y)s(p)s(r) + c(y)c(r) & s(y)s(p)c(r) - c(y)s(r) \\ -s(p) & c(p)s(r) & c(p)c(r) \end{bmatrix} \tag{3.1}$$

The Rift's coordinate system can be seen in figure 3.3. There is a possibility that this system doesn't conform with the one being used by the models and the whole environment created in OpenAR. In this case, the head rotation matrix must be transformed to abide the rest of the experience. It is just a matter of applying the desired transformation to the head matrix.

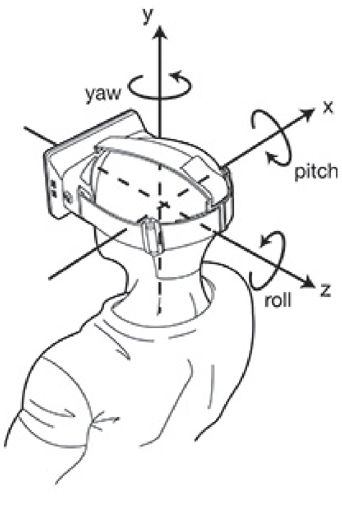

FIGURE 3.3: Rift axis system

With a rotation matrix conformed to the defined standard, a front and an up vectors are now needed to position the virtual camera in OpenAR keeping in mind that the front vector also includes the desired translation. These two vectors are defined as needed, multiplied by the head's rotation matrix and passed on to the virtual camera correctly positioning it.

3.3 Kinect, OpenNI and NITE

Kinect integration is done by using OpenNI and NITE. These two libraries handle the complicated process of acquiring the RGB image and depth map provided by the Kinect. Those two data sources are then automatically matched to a certain set of bone's/joint's transformations. Though it is said that the Kinect is able to track up to 20 joints, our experience showed that some of those joints usually had inconsistent or almost constant values. With this in mind, it was decided to discard those joints (left and right wrists, ankles, head, neck, hands and feet) and track only the reliable ones: elbows, shoulders, hips, knees and torso. The reason for discarding the head and neck is not only the unreliable nature of the values returned by the Kinect for those joints but also the fact that the values from the Rift's gyroscope will be used to orient these joints as will be explained later on. The feet and hands' joint information are not needed because their position is given by the rotations of the joints that precede it. All the joints traceable through the Kinect are presented in figure 3.4, the ones in blue are the ones being used in this work.

One of the problems when using Kinect is the noise noticed in the depth camera. This noise causes a noticeable jitter on the avatar, especially when experienced from a first person perspective. Thankfully OpenNI has a built-in noise-cancelling filter that can be configured to balance between introduced delay and noise filtering. Delay and jitter are both incredibly damaging to the immersive experience so several tests were made to find an acceptable trade-off. A class serving as a bridge between Kinect's framework and OpenAR was created. This class has all the methods for collecting the joints transformations in a format compliant with the rest of the program's pipeline and automatically initiates and configures the Kinect sensor

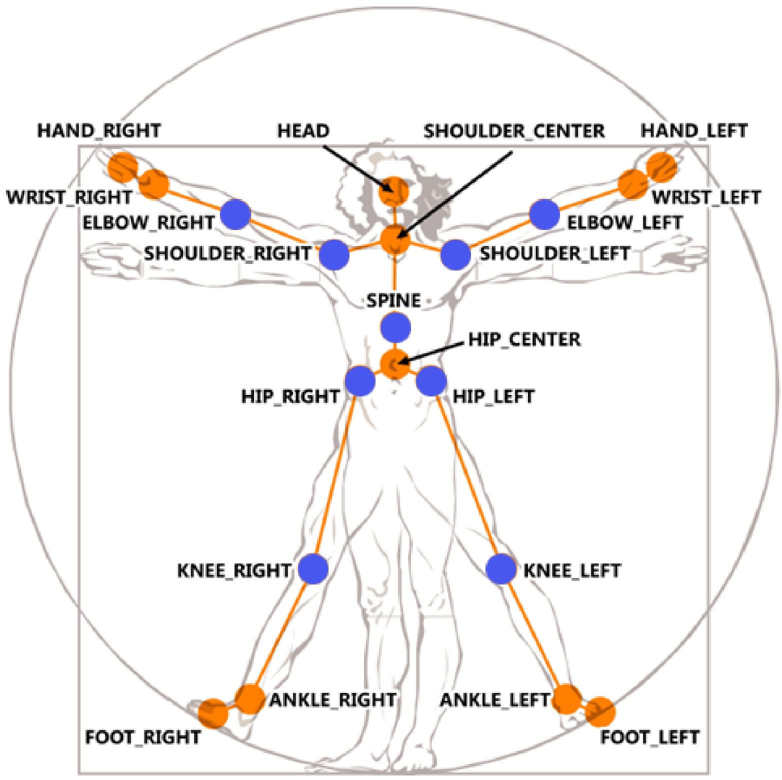

FIGURE 3.4: Kinect's skeleton tracking nodes (nodes being used are presented in blue)

3.4 Animation Engine

KinectAnimatedBody was the class developed to hande the real-time animation of the virtual avatar with the data collected from the Kinect. To understand how this animation is achieved one must first understand the concept of model rigging.

After the model's mesh is completed it has to be rigged. Rigging is the process of creating virtual bones and choosing the weight that those bones have on each of the mesh's vertices so that, by moving a certain bone, all the vertices influenced by it also move accordingly (like by moving our arm we are moving parts of our skin). It is important to notice that a single vertex can be influenced by several bones and this fact helps distorting the mesh in a less rigid manner creating smoother animations. This mesh or skin deformations is performed, not by the animation system, but by the shader in a later stage of the graphics computation pipeline. The complete set of bone transformations is passed on to the shader (the code that runs on the graphics card) and each of the mesh's vertex is multiplied by the transformation of all the bones that influence and their respective weights.

These bones are organized on a parent-child hierarchy, meaning that by applying a certain transformation to a parent node, that same transformation will also be applied to all its children and that each bone's transformation is given relatively to its parent. None of this works all that differently from our structure as humans. When we move our shoulder we are also moving our elbow and hand, even though our elbow and hand didn't move relatively to our shoulder. This implies that the transformation of each bone is given in its own coordinate's system, resulting from the collective transformations of all the previous bones in the hierarchy.

Another detail to take into consideration is the way that each bone is oriented. Modelling and animation engines like Maya and Blender always place the bones y-axis along the virtual bone from the base to the tip. The standard structure of the skeleton used in all our models can be seen in figure 3.5. This was the skeleton used to rig the models and possesses the same number of nodes bones as the ones tracked by the Kinect (shown in figure 3.4) plus the neck that, as previously mentioned, uses data from the Rift's head tracker.

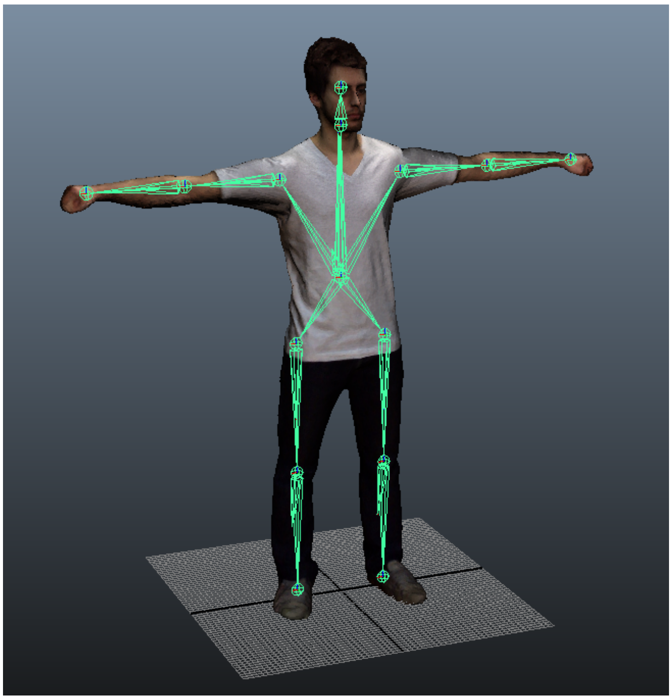

Figure 3.5: Bone structure created in Maya

The process of creating a regular animation requires the modeller to first rig the mesh as said above and then generating a number of keyframes. These keyframes are a map of the positions and orientations of all the model's bones at a given time. What the animation engine does when animating the model is to collect those keyframes from the animation file and interpolate the bone's transformation between animation keyframes. By doing this, the workload of the modeller is greatly reduced by not having to manually place every bone in its proper place for every frame of the animation.

The animation engine required to achieve our Kinect animated avatar is actually simpler than the one needed for full animation. There's no need to create keyframes or interpolate the bone's transformations because, in each frame, the updated bone's transformations can be retrieved from the Kinect and transformed to abide the model bone structure and hierarchy.

Unlike what was said above regarding virtual bone's axis in Maya and Blender, the Kinect sensor captures the transformation (position and orientation) of each bone in a global axis system having no notion of hierarchy whatsoever. In this case the position captured by the Kinect can be discarded since we are trying to animate a model with a fixed bone structure. There is no interest in translating the bones from its original position since this is already achieved by applying the bone's rotations extracted from the Kinect and respecting the hierarchy.

This difference between the Kinect's and the model's coordinate systems generates a problem that must be addressed for proper avatar animation. This means that, for each bone, the corresponding Kinect transformation matrix has to be transformed from the Kinect's axis system to the model bone's own axis. This is achieved by a function that runs through all bone's starting from the root (first node in the hierarchy) and recursively jumping to each children while keeping the axis transformations applied until that bone. By doing this, in every node, the function has a matrix composed of all the applied rotations. At this point, it's just a matter of multiplying the adequate Kinect matrix by this transformation. A more detailed description of this recursive function is shown in algorithm 1.

```
nodePointer ← the pointer to the bone (starts by being the root);
parentTransformation ← the final transformation of the parent (starts as
identity);
parentOrientation ← the standard orientation of the parent (starts as identity);
nodeOrientation = getNodeOrientation;
nodeOrientation = parentNodeOrientation * nodeOrientation;
nodeTransformation = getNodeTransformation;
nodeTransformation = parentTransformation * nodeTransformation;
if nodeHasRotation then
    kinectTransformation = getKinectTransfomation;
    kinectTransfomation = kinectTransformation * nodeOrienation;
    merge(nodeTransformation,kinectTransformation);
end
for nodePointer→children do
    readNodeHierarchy(childrenPointer, nodeTransformation, nodeOrientation);
end
```

Algorithm 1: readNodeHierarchy: Recursive function that runs the bone's hierarchy

3.5 Introducing Scaling in the Animation Engine

At a certain time of the development process and following the trend of applying immersive VR systems to the fields of neuropsychology and, specifically, the treatment of eating disorders [?] the interest to scale parts of the virtual avatar arose. The ability to scale the belly or any other body part of the avatar to make it look thinner or larger and putting the user through the eyes of his/her modified self can have great impact on the sense of self-awareness.

Two methods were developed to achieve this effect and both required the addition of one more step when running through the bone's hierarchy. The first is just a set of scaling matrices that can be set at any time during course of the program and are then applied to the correct bones. This method allows for some interesting applications since the scaling of any bone can be altered in runtime, these applications will be discussed further later on.

The second method is more complex and resembles the one described on section 3.4 for full animation. This method requires the mesh to be animated in Blender or Maya. If some scaling animation is detected in the model file, that set of scaling matrices is applied to the model in the same step as the first method. The difference between this animation engine and a regular one is that the animation may not follow the normal order of the set keyframes. In an experimental scenario this can be useful so that the supervising expert can choose the preset animation frame at-will. The idea is to create an animation that morphs the same avatar through several different levels of body fat so that the user can experience those body types. When a certain keyframe is selected the engine just applies that set of transformations to the bones correctly modifying the avatar.

Unfortunately, the introduction of a scaling system and the possibility of transforming the original mesh generates a problem where parts of the model's mesh can sometimes overlay other parts of that same mesh. For example, if the user scales his/her virtual model's belly and then touches his/her real belly, the virtual hand will appear to be inside his/her virtual enlarged belly. Some users reported this occurrence as being damaging to the systems SoE and so this problem could not be ignored.

3.6 Solving the Problem of Mesh Overlaying

From the point of view of the user, the mesh overlaying problem only presents itself when he/she is trying to touch his/her avatar's belly so this is the circumstance that is going to be addressed from now on.

The solution to the problem of mesh overlaying involves three steps. The first step is detecting that the virtual avatar's hand is overlaying another part of the avatar's mesh. The second step involves the calculus of a new appropriate position for the hand that doesn't overlay the mesh. The third step involves placing the hand in the calculated correct position. This third step is the most complex one because in order to change the hand's positions new rotations for the shoulder and elbow joints have to be calculated.

The approach used to solve the first part of this problem, and given the fact that the avatar joints' positions are known at all times, was to create a cylinder placed along the torso and spanning from the waist through the shoulder's line. The hand's positions are tested against this cylinder. If any of the hands is indeed inside this cylinder, then the hand is considered to be overlaying the mesh and that hand must go through steps two and three.

The algorithm to test if a given point is inside the cylinder is described in 2. This algorithm receives the center points of the cylinder's top and bottom caps, the point to test against the cylinder, the length and radius of the cylinder and returns -1 if the point is outside or the distance to the axis of the cylinder if the point lies inside it.

```
cylBot ← waist position;
cylTop ← neck position;
lengthSq ← neck-waist distance squared;
radSq ← defined cylinder radius squared;
testPoint ← hand position;
// set cylBot as the origin
d = cylBot - cylTop;
pd = testPoint - cylBot;
// dot product between test point and cylinder bottom
dot = pd.d;
// check if test point is between cylinder caps
if dot < 0 or dot > lengthSq then
    // if test point is outside
    return -1;
else
    // if test point is inside compute distance to cylinder axis
    dsq = pd^2 - dot^2/lengthSq ;
    if dsq > radSq then
        // if distance to axis is greater then radius
        return -1;
    else
        // if distance to axis is smaller then radius
        return dsq;
    end
end
```

Algorithm 2: Test if a point lies inside a cylinder

It is important to note that, when the avatar's belly is scaled, this cylinder is also scaled accordingly. This way, the volume considered as illegal hand space is always adjusted to the avatar's size.

The second step involves calculating a new and legal hand position. Taking into account the fact that an adjusted cylinder is already correctly placed around the avatar's waist, the new hand position is assumed to be the closest point in that cylinder's surface to the actual hand position. This way, if the user moves the virtual hand inside illegal space the hand will move through the cylinder's surface accordingly.

To calculate this legal hand position the current hand height is kept so the problem becomes two-dimensional. The vector from the center of the cylinder slice with the same height as the hand is calculated. The module of this vector will be smaller than the cylinder radius, otherwise the hand wouldn't be in an illegal position. This vector is then scaled to match the cylinder's radius and again applied to the center of the cylinder slice. This will give us the desired position for the hand.

To solve the problem involving the third step it is important to consider certain aspects: the forearm and upperarm sizes are constant, the hand, shoulder and elbow positions are known at all times and the elbow is a hinge. Because of the fact that the elbow is a hinge, the elbow bend angle can only depend on the distance between the shoulder and the desired hand position. With all this in mind we can assume the problem of calculating the elbow bend angle as a 2D problem and use the following equation plugging the forearm size as S_2, the upperarm size as S_3 and the distance between the shoulder and the desired hand position as S_1.

$$\text{ElbowBend} = \arccos\left(\frac{S_2^2 + S_3^2 - S_1^2}{2 * S_2 * S_3}\right) \tag{3.2}$$

This equation always returns the angle opposite to the side 1 (S_1). Now the angle elbow bend angle is calculated but for the animation engine to correctly apply that rotation a complete rotation matrix is needed. The way the human elbow works is that the elbow bend is always applied around an axis normal to the plane that contains the hand, the elbow and the shoulder joints. The cross product between the forearm and the upperarm vectors gives this axis and to get the desired complete rotation matrix a rotation of ElbowBend around that calculated normal axis is performed.

Now the forearm already has the desired rotation but the shoulder also has to be rotated to accommodate the elbow bend. The first step is to point the upperarm directly at the desired hand position. This was accomplished by calculating a quaternion based on two vectors ($\vec{u}$ and $\vec{v}$). The first vector ($\vec{u}$) is the standard arm position (usually aligned with the X or Y axis) the second vector ($\vec{v}$) is the one that starts at the shoulder position and ends at the desired hand position. By first normalizing these two vectors and then making:

$$\vec{w} = \vec{u} \times \vec{v} \tag{3.3}$$

$$\mathbf{q} = (1 + u_x v_x + u_y v_y + u_z v_z, w_x, w_y, w_z) \tag{3.4}$$

This gives a quaternion $\mathbf{q}$ representing the rotation needed to transform the shoulder from its original orientation to pointing directly at the desired hand position. This quaternion must now be transformed into a rotation matrix to conform with the other transformations. This is done with the following equation assuming $\mathbf{q} = (x, y, z, w)$.

$$\mathbf{R} = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2zw & 2xz + 2yw \\ 2xy + 2zw & 1 - 2x^2 - 2z^2 & 2yz - 2xw \\ 2xz - 2yw & 2yz + 2xw & 1 - 2x^2 - 2y^2 \end{bmatrix} \tag{3.5}$$

This shoulder rotation matrix still doesn't take into account the elbow bend and so further calculations are needed in order to get the shoulder bend to accommodate the elbow bend. The same equation used to get the elbow bend is used but with S_1 now being the forearm size, S_2 the distance between the desired hand position and the shoulder and S_3 the upperarm size. This shoulder bend is then transformed into a matrix by rotating around the same axis as was the elbow bend (the elbow normal) which gives the shoulder bend matrix. Lastly, the shoulder bend matrix is multiplied by the first shoulder rotation matrix giving the complete shoulder rotation needed to reach the desired point.

3.7 System Loop and Other Details

At this point we have an avatar correctly animated by the user's actions captured through the Kinect sensor but a few other aspects need to be addressed.

One of which is the avatar placement on the scene. Since until now only joint rotations are being used, the avatar still doesn't mimic the user's position in real space. A step was introduced in the system loop where the distance between the Kinect and the root node (torso) is measured and applied. Since the avatar object complies with the standard defined by OpenAR, it can be moved with the engine's built in functions to move objects and, since all the virtual environments will be created on a 1:1 scale with the real space, the Kinect to user distance can be directly applied.

The scenario model also has to be properly positioned according to the real space around the user, the Kinect's position and the avatar's height. Starting by the avatar's height, it is important to note that it is desired that the avatar's feet always touch the ground. To this effect, and assuming that the avatar's root is always the torso, the distance between the torso and lowest foot is measured. This distance is inversely applied to the entire scenario effectively keeping the floor below the avatar's feet. This however prevents the user from being able to jump since, if both feet are lifted from the ground, the scenario will still track the lowest foot.

To sync the virtual and the real worlds a simple configuration method was introduced. There's a key that, when pressed, defines the user's and the avatar's positions as the center of the virtual world. The idea is to ask the user to stand in the center of the real space and then configure the virtual scenario so that both are aligned. With this simple setup, we extract the rigid transformation between the Kinect and the virtual world referential. Then, and keeping in mind that by now the user and the user's avatar are in the same position relative to the Kinect's referential, the user's bone rotations can be extracted (in relation to the Kinects referential) and applied to the avatar. The avatar is now correctly placed and animated in the virtual scene. The virtual camera now has to be placed in the correct position (the avatar's eyes) to provide the desired egocentric first person view. The Rift has its transformation to the virtual world but, knowing that the Rift is being worn in the user's head, it is possible to apply to the Rift (and the virtual camera) the same transformation that is applied to the avatar's head. This closes the loop of transformations that allow us to synchronize the virtual an real worlds, place the avatar on the scene and place the virtual camera on the avatar's head as presented in figure 3.6.

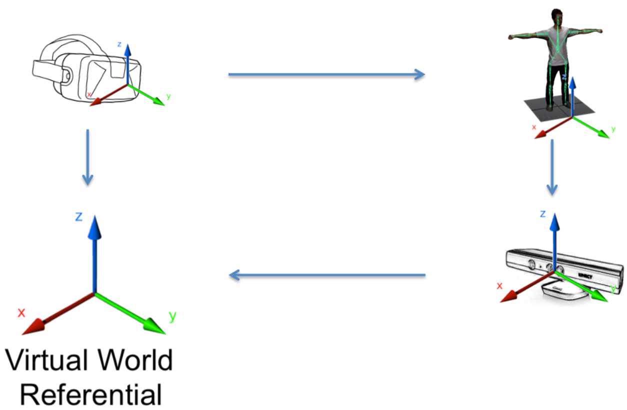

FIGURE 3.6: Transformations between integrated systems

As was mentioned before, the Kinect's head/neck rotations aren't reliable. Since a HMD equipped with gyroscope sensors is being used, it was decided that the neck rotation would be acquired from the Rift instead of the Kinect. When the joints rotations are passed on to the avatar object the neck rotation is replaced by the rotation extracted from the Rift (correctly transformed).

To sum up the process, a list of all steps performed during a regular system loop is presented below.

1. Retrieve a new set of bone transformations from the Kinect

2. Retrieve head rotation from the Rift and transform it
3. Replace Kinect's neck rotation by the one from the Rift
4. Run through the model's hierarchy and apply the new set of bone transformations
5. Check if the model's hands are overlaying the belly zone
    (a) Calculate new legal hand position
    (b) Calculate elbow and shoulder transformations to reach that position
    (c) Run the model's hierarchy to apply the new transformations
6. Place the model on the VE according to user-Kinect distance
7. Place the virtual camera on the head node of the model
8. Render scene

Chapter 4

Demos and Case Studies

4.1 The Mirror

After the base system functionality was completed some demos were developed. The first one, and the one that served as a starting point to many others was the mirror. The idea behind this demo was, as the name implies, the recreation of a mirror where the user would be able to closely inspect both his/her virtual body and the virtual body's reflection. Moving in front of a mirror while being able to see your actions mirrored is a familiar experience that provides a great sense of embodiment and immersion. To improve this effect, and keeping in mind that the user has the Rift put on during the course of the experience, it was important to create a virtual room with the same dimensions as the real space that the user would be traversing. A square of 2 by 2 meters was used for this experiment with walls on all sides but one and a virtual room with the same size and scale was created. The virtual mirror was placed on the side with no real wall (also being the side where the Kinect is placed). With this configuration, the user can actually lean on or touch a wall with the avatar perfectly recreating those actions in the virtual space. This conformity between virtual and real spaces greatly improve the system's immersion.

On standard conditions, the mirroring effect could be achieved by placing a virtual camera on an appropriate place and then applying the rendered texture of that camera to the virtual wall serving as mirror. This is not possible in this case though because the avatar directly controlled by the user doesn't have a head since the virtual camera that renders the scene to the Rift is placed on the same place where the avatar's head would be. With this in mind, a small and unnoticeable trick was applied. Instead of having one room and one avatar on the scene, two rooms and two avatars are actually placed. The two rooms are perfectly mirrored and placed right next to each other and two avatars, one without a head controlled by the user and a complete one serving as the reflection are also placed on the appropriate rooms. The joint rotation matrices applied to the reflected avatar are the same matrices applied to the user's avatar but multiplied by a symmetry matrix correctly mimicking the effect of a real mirror.

The following image shows an user inside the mirror room.

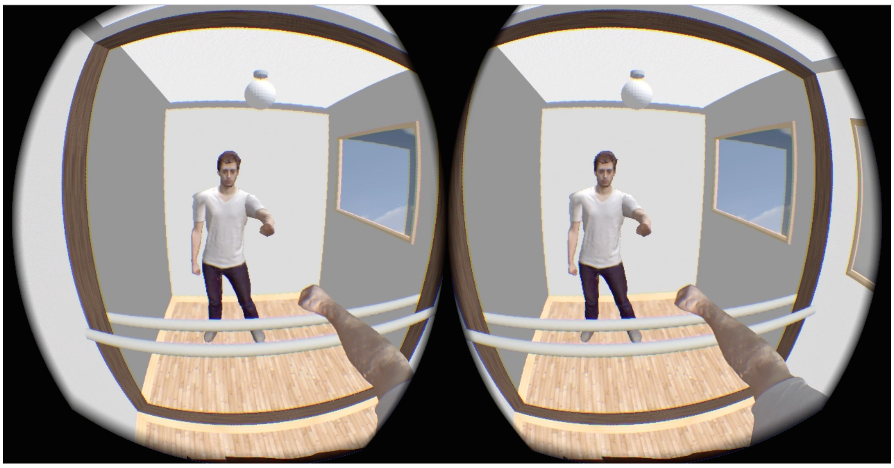

Figure 4.1: The Mirror

4.2 The Guillotine

The guillotine was a demo developed to be exposed at exPat 2015. This conference took place in Azores and had in mind the demonstration of interesting technologies for education and learning. Using the immersive system that was developed and one other peripheral capable of acquiring some biological signals, a proposal was made with the intention of teaching the physiological effects of different stimuli. The idea was to put the user through a set of experiences while monitoring his/her biological data and then, after the experience was completed, try to correlate certain peaks with the events that occurred.

The guillotine consisted on the same environment as the mirror with the addition of a traumatic or scary event. At a certain point of the experience the user is asked to slowly reach for the mirror. When the program detects the avatar's proximity with the mirror several events are started. A bloody guillotine falls from the ceiling effectively severing the avatar's hand and blood starts spouting from the end of the arm. This is all accompanied by appropriate blade and squishy sound effects. While this happens, an electrodermal activity (EDA) sensor is attached to the user's hand and by plotting the collected data it can clearly be seen that a peak in EDA data occurs seconds after the falling of the guillotine. This small delay (usually 2/3 seconds) is natural of the signal being read, EDA being a delayed physiological response.

Three examples of the acquired EDA responses are shown in figure 4.2. As can clearly be seen, users had significant spikes in EDA data following the fall of the guillotine. This can be seen as an indicator of the immersiveness of the experience.

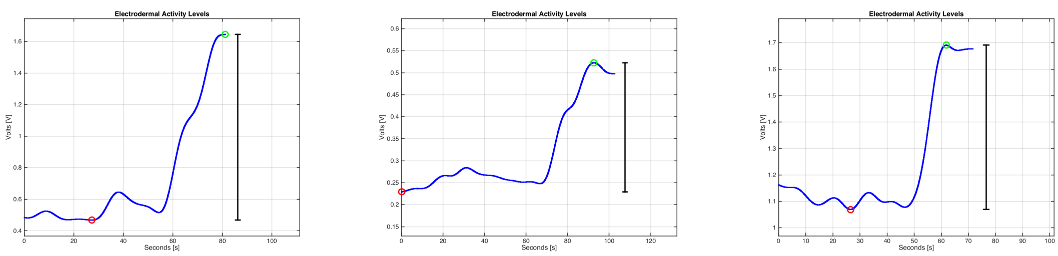

FIGURE 4.2: EDA responses for 3 different users

The following figure shows this experience's sequence of events, from the user approaching the mirror to the falling of the guillotine and the severing of the hand. Only the image shown to the left eye is presented.

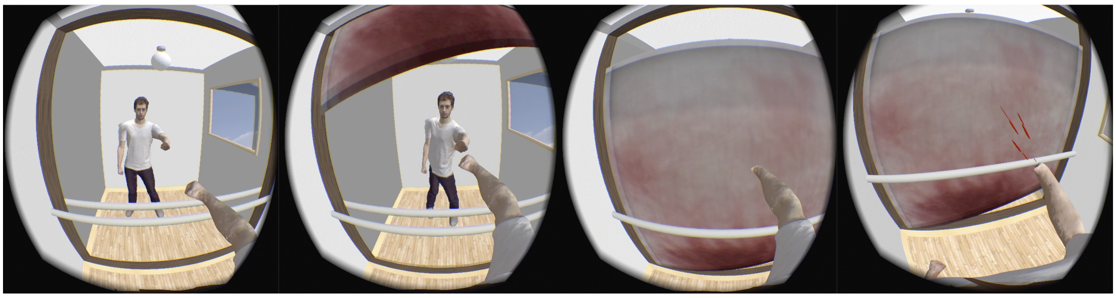

FIGURE 4.3: The Guillotine sequence of events, from left to right

4.3 The Body Editor

This demo was the one that made use of all the implemented features. Its creation had in mind the clinical environment of psychologists treating eating disorders and body image dysfunctions. The scenario is the same as the one used in The Mirror and The Guillotine but close to the virtual mirror a set of buttons were placed. Each of these buttons scales a different body part (torso, legs, arms, head and hands). If a certain button is pressed with the left hand, a negative scaling is applied shrinking the respective body part. If pressed with the right hand the opposite happens. The user can interact with these buttons and observe the effects in real-time both on his/her own avatar and on the avatar's reflection. After some tests we realized that our bone structure wasn't fit for this experiment because scaling the torso would result in the avatar looking more muscular instead of fatter. With this in mind another joint was added to the bone structure. This joint we called 'belly joint' influences only the vertices belonging to the avatar's lower abdomen and is this joint that is scaled when the torso button is pressed. With this we achieved the desired effect of being able to make the avatar look thinner or fatter.

People with body dysmorphic disorder or other diseases that affect the perception of one's body are usually unable to realize how they really are. Psychologists usually apply treatments based on observing their reflection or drawing themselves. This demo presents a novel approach to these treatments serving as a more controlled and interactive experience for the patient.

The following figure shows the same avatar with three different scaling configurations. The first image is the regular one, the second shows an enlarged belly and the third presents a thinner belly and thinner arms. It's important to note that only the image corresponding to the left eye is being shown.

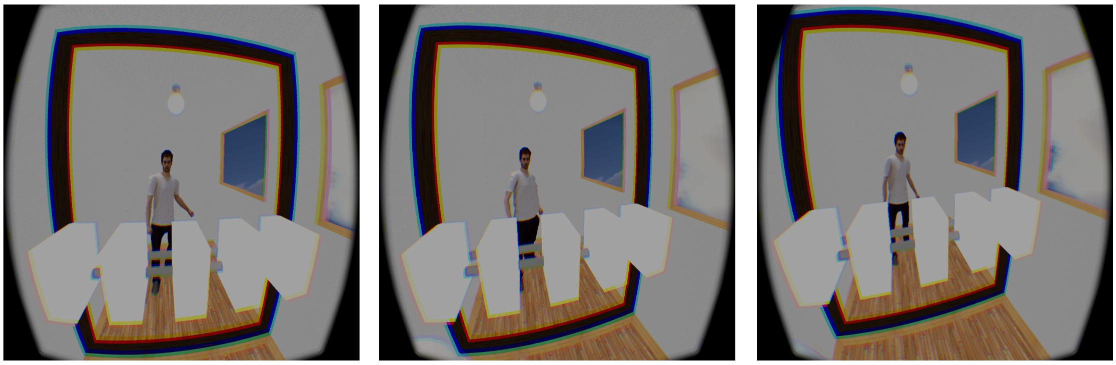

Figure 4.4: Body Editor example

4.4 The Submarine

The submarine was a side project that resulted from a partnership between our laboratory and the Department of Computer Science and Engineering, University of Castellón, Spain. They had a submarine that was operated through a set of monitors as can be seen in figure 4.5. They thought that this was not an intuitive approach to teleoperation and that a VR based system could be more natural and intuitive. In addition, the complex control interfaces, which require skilled pilots, present several drawbacks to the user like cognitive fatigue and high stress inherent to master/slave control architectures.

We set to build a better interface for this system. We wanted to make the control of the robot as natural as possible so we chose to create VR windshield. This virtual windshield is placed around the user and contains any number of gauges and dials depending on the necessity for the current operation. My contribution to this project was the creation of a framework for easy creation and placement of these indicators. The types of indicators created are presented in figure 4.6. From left to right: text boxes, warning lights, straight analog dial and round analog dial. An example of their usage is shown in 4.7.

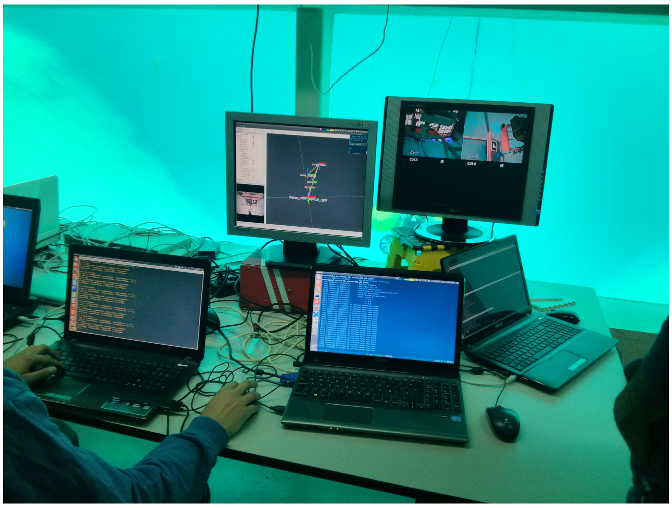

FIGURE 4.5: Submarine control and monitoring old interface

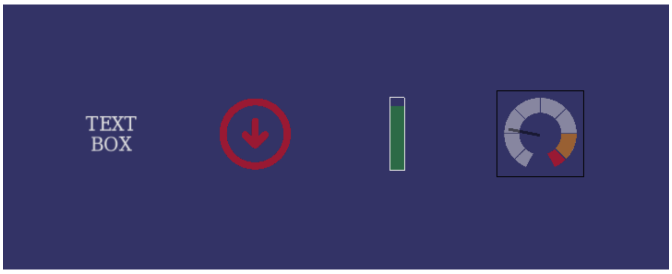

FIGURE 4.6: Developed interface components

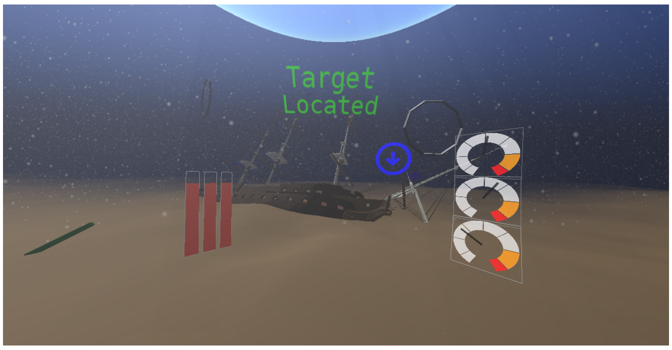

FIGURE 4.7: Interface Example

All of this was done in C++ and Open Scene Graph (OSG) and is just another example of the many applications that immersive systems have. To evaluate the effectiveness of this new interface a study was conducted where we asked 30 users to test both the standard method (several screens, non-immersive) and the new one (VR-based, immersive). The same data was presented in both methods.

As can be seen in figures 4.8 and 4.9, the vast majority of the users described the new interface as very helpful and the inclusion of VR as a great improvement.

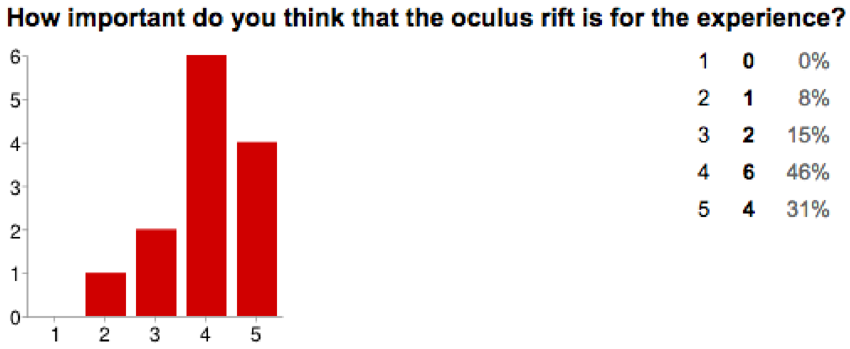

Figure 4.8: Rift importance from user questionnaire comparing to old interface

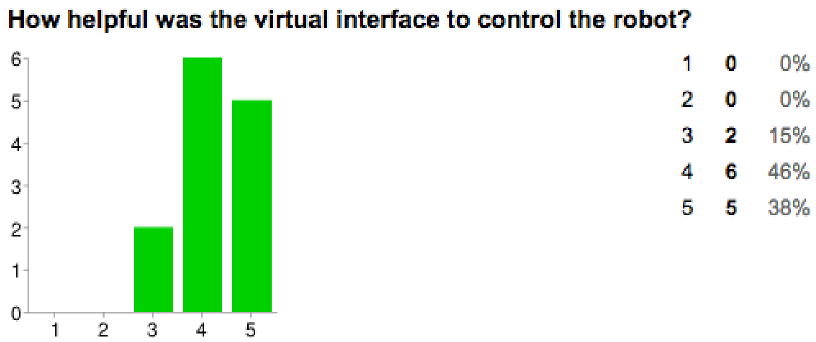

Figure 4.9: Helpfulness of the virtual interface from user questionnaire

Chapter 5

Conclusion

This work was centred on the development of the necessary support for virtually transporting a user to a synthetic environment. To this end, a set of functionalities and applications were developed. Support for Rift and Kinect was added to the existing framework and the addition of realistic user 3D models was performed. Whenever possible the results of the applications were evaluated using task-related performance parameters, user questionnaires, or by observation, as last resort.

In our experiences users were able to easily interact with the Body Editor and the buttons placed on the scene implying that they had a sense of body ownership and recognized the volume occupied by their avatars. The physiological responses stimulated by the traumatic experience presented on The Guillotine revealed that users were indeed immersed in this virtual environment and were able to recognize threats to the avatar as threats to themselves, implying that they recognized the avatar as themselves or at least as an extension. The Submarine was a successful experiment in adding an immersive dimension to teleoperation being reported as very important to its manipulation. Another great highlight was the interest of the psychologist community on the experience created in The Body Editor.

Given the success of the various experiments, one can say that the objective of transporting an user to a virtual environment was fulfilled.

5.1 Further Work

Even though several parts of this work have already been scientifically recognized, several other features could be implemented. Other feedback systems could be added to improve its extensiveness. Hand tracking would be advantageous and would enable further interaction between the user and the virtual environment. Some sort of haptic feedback could be integrated to create even more immersive experiences. The inclusion of spatial sound could also be a fruitful line of work to follow.

Apart from these improvements, the most important work to follow would be the conduction of a case study in a clinical environment using the Body Editor. Testing the system in a real life scenario and with real patients with the help of an expert would be a challenge that could produce interesting results.

Bibliography

[16] Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for Industrial Applications*, pages 282–292. International Society for Optics and Photonics, 1995.

[1] Morton Heilig. The cinema of the future. *Translated by Uri Feldman. In Multimedia: From Wagner to Virtual Reality. Edited by Randall Packer and Ken Jordan. Expanded ed. New York: WW Norton*, pages 239–251, 2002.

[2] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM.

[3] Barbara O Rothbaum, Larry F Hodges, David Ready, Ken Graap, and Renato D Alarcon. Virtual reality exposure therapy for vietnam veterans with posttraumatic stress disorder. *Journal of Clinical Psychiatry*, 2001.

[4] Joann Difede and Hunter G Hoffman. Virtual reality exposure therapy for world trade center post-traumatic stress disorder: A case report. *Cyberpsychology & Behavior*, 5(6):529–535, 2002.

[5] Thomas D Parsons and Albert A Rizzo. Affective outcomes of virtual reality exposure therapy for anxiety and specific phobias: A meta-analysis. *Journal of behavior therapy and experimental psychiatry*, 39(3):250–261, 2008.

[6] Craig D Murray, Stephen Pettifer, Toby Howard, Emma L Patchick, Fabrice Caillette, Jai Kulkarni, and Candy Bamford. The treatment of phantom

limb pain using immersive virtual reality: three case studies. *Disability and rehabilitation*, 29(18):1465–1469, 2007.

[7] Giuseppe Riva. Virtual reality in psychotherapy: review. *Cyberpsychology & behavior*, 8(3):220–230, 2005.

[8] Larry F Hodges, Rob Kooper, Thomas C Meyer, Barbara O Rothbaum, Dan Opdyke, Johannes J De Graaff, James S Williford, and Max M North. Virtual environments for treating the fear of heights. *Computer*, 28(7):27–34, 1995.

[9] Paul MG Emmelkamp, Mary Bruynzeel, Leonie Drost, and Charles AP G van der Mast. Virtual reality treatment in acrophobia: a comparison with exposure in vivo. *CyberPsychology & Behavior*, 4(3):335–339, 2001.

[10] Azucena Garcia-Palacios, H Hoffman, A Carlin, TA u Furness, and C Botella. Virtual reality in the treatment of spider phobia: a controlled study. *Behaviour research and therapy*, 40(9):983–993, 2002.

[11] C Botella, RM Baños, C Perpina, H Villa, M u Alcaniz, and A Rey. Virtual reality treatment of claustrophobia: a case report. *Behaviour research and therapy*, 36(2):239–246, 1998.

[12] F Vincelli, YH Choi, E Molinari, BK Wiederhold, and G Riva. A vr-based multicomponent treatment for panic disorders with agoraphobia. *Studies in health technology and informatics*, 81:544–550, 2000.

[13] Giuseppe Riva, Monica Bacchetta, Margherita Baruffi, Silvia Rinaldi, and Enrico Molinari. Experiential cognitive therapy: A vr based approach for the assessment and treatment of eating disorders. 1998.

[14] Giuseppe Riva, Monica Bacchetta, Margherita Baruffi, Giovanna Cirillo, and Enrico Molinari. Virtual reality environment for body image modification: A multidimensional therapy for the treatment of body image in obesity and related pathologies. *CyberPsychology & Behavior*, 3(3):421–431, 2000.

[15] Bernhard Spanlang, Jean-Marie Normand, David Borland, Konstantina Kilteni, Elias Giannopoulos, Ausias Pomes, Mar Gonzalez-Franco, Daniel

PÃĨrez Marcos, Jorge Arroyo Palacios, Xavier Navarro Muncunill, and Mel Slater. How to build an embodiment lab: Achieving body representation illusions in virtual reality. *Frontiers in Robotics and AI*, 1(9), 2014. ISSN 2296-9144.

[17] M Slater, A Steed, and M Usoh. Being there together: Experiments on presence in virtual environments (1990s). Technical report, Technical Report, Department of Computer Science, University College London, UK, 2013.

[18] Betty J. Mohler, Sarah H. Creem-Regehr, William B. Thompson, and Heinrich H. Bülthoff. The effect of viewing a self-avatar on distance judgments in an hmd-based virtual environment. *Presence: Teleoper. Virtual Environ.*, 19 (3):230–242, June 2010. ISSN 1054-7460.

[19] Konstantina Kilteni, Raphaela Groten, and Mel Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2015/07/09 2012.

[20] Mar González-Franco, Tabitha C Peck, Antoni Rodríguez-Fornells, and Mel Slater. A threat to a virtual hand elicits motor cortex activation. *Experimental brain research*, 232(3):875–887, 2014.

[21] Stéphane Bouchard, Julie St-Jacques, Geneviève Robillard, and Patrice Renaud. Anxiety increases the feeling of presence in virtual reality. *Presence: Teleoperators and Virtual Environments*, 17(4):376–391, 2008.

[] C Perpiñá, C Botella, and RM Baños. Virtual reality in eating disorders. *European Eating Disorders Review*, 11(3):261–278, 2003.